
HTML NOTES

I'm going to give you a few simple lessons on how to make a Web Page. I must warn you though, this is for "all wet behind ears" Newbies. If you're at all experienced at this sort of thing, you'll probably find this tutorial a bit of a yawner.

You'll be happy to learn that it's really pretty simple. The basic idea is this... A web page is nothing more than a file, a HTML file to be exact. It's called HTML because web page documents have the file extension `.html` or `.htm`. HTML stands for **H**yper **T**ext **M**ark-up **L**anguage. (If you are unclear about this file extension stuff, then you really *are* newbie!! Take a [quick detour](#) for a few ramblings on the subject).

Let's get started. First, if you have any of them fancy HTML editors and have an inkling to break em in now... forget it. The worst way to learn is to use one of those things. (Although there are a few that you will find helpful, they'll only help you once you learn the basics, so don't even worry about it now.) What's the best way to learn HTML?? Notepad. I know, I know, you got this 9 megabyte Wunder Wizzard that says it's gonna make putting up a web page as easy as scratching your head. Just trust me on this one. K? K. *(If you're just looking for a super easy way to pop up a simple page then I'll save you some time and point you to Web Wizard)*

Another tool you'll need is a *browser* to view your web pages. Netscape Navigator is what I strongly recommend. If you're using something else, it should be OK for most things, but you may be somewhat limited. If you're using something provided by America Online, CompuServe, or one of those Internet in a Box packages, you're on your own because I have no idea what they give you to work with.

With that out of the way I can say with confidence that you are less than 5 minutes away from making your first web page! So.. off to lesson 1.

Just like 'follow the bouncing ball', power up Notepad and start with this...

```
<HTML>
</HTML>
```

Each one of those is called a tag. There is a starting tag and a closing tag. To make a closing tag just add a / to the starting tag. Most, but not all tags have a closing tag.

Think of tags as talking to the browser, or better yet giving it instructions. What you have just told the browser is 'this is the start of a HTML document' (`<HTML>`) and 'this is the end of a HTML document' (`</HTML>`).

Now we need to put some stuff into it.

Every HTML document needs a pair of HEAD tags.

```
<HTML>
<HEAD>
</HEAD>
</HTML>
```

The only thing we have to concern ourselves with in the HEAD tags (for now) are the TITLE tags.

```
<HTML>
<HEAD>
<TITLE></TITLE>
```

```
</HEAD>
</HTML>
```


And the bulk of the page is going to be within the BODY tags.

```
<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Oh, and one more thing, give your document a title.

```
<HTML>
<HEAD>
<TITLE>Welcome to IT Kids!</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Now save it, not as a text document, but as a html document. Save it as `page1.html` in a new folder somewhere (Win3.x users save it as `page1.htm`). If yer a little fuzzy about how to do this then here's what you do (Win95)... In your Notepad window click File then Save as.

You will be presented with a dialog box. Make a new folder by clicking this ->  Name it whatever you want. Then double click on it to open it up. Where it says File name: type in `page1.html` Where it says Save as type: make sure it says All Files (*.*) Hit return and you're done!

Congratulations! You are the proud parent of a fully functional Web Page! You could upload it to a server and the whole world can see your creation! If you are using Netscape, the file you made should look something like this...



(if your Netscape icon is a little different, it's no biggie)

You can double click on it now and see the [results of your handiwork](#).

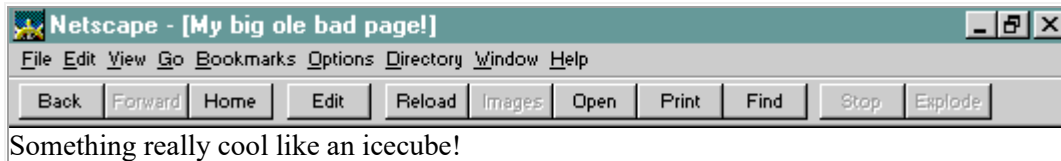
Unfortunately, as you can see, the page is just a little bit blank (but it is still a legitimate HTML document!). Next order of business is to start putting some neato stuff in your page.

The best way to use this tutorial is to run Notepad and two instances of Netscape. One Netscape window containing this tutorial and the other containing your new page. Just toggle between the three windows. If you are looking at this lesson in Netscape right now, you can open a second instance of Netscape in one of two ways- 1) Find the icon of the html file you just made (`page1.html`) and double click on it. Or- 2) In Netscape, click on File/Open File In Browser and browse to the file (`page1.html`).

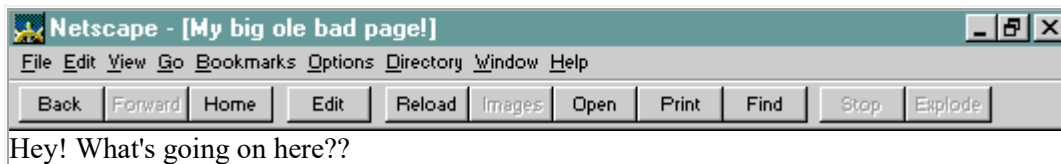
Two quick points before we go on to lesson 2. One, what you made is a skeleton HTML document. This is the minimum required information for a web document and all web documents should contain these basic components. And two, the document title is what appears at the very top of the browser window.

I think we'll start by learning something about the way a browser works.
First an example...

```
<BODY BGCOLOR="#FFFFFF">
Something really cool
like an icecube!
</BODY>
```

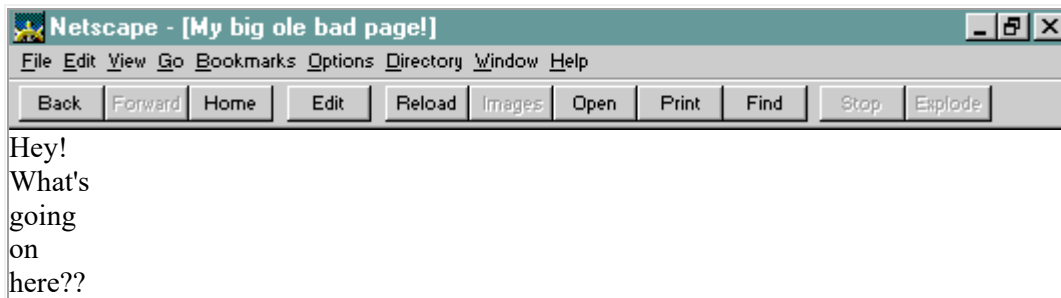


```
<BODY BGCOLOR="#FFFFFF">
Hey!
What's
going
on
here??
</BODY>
```



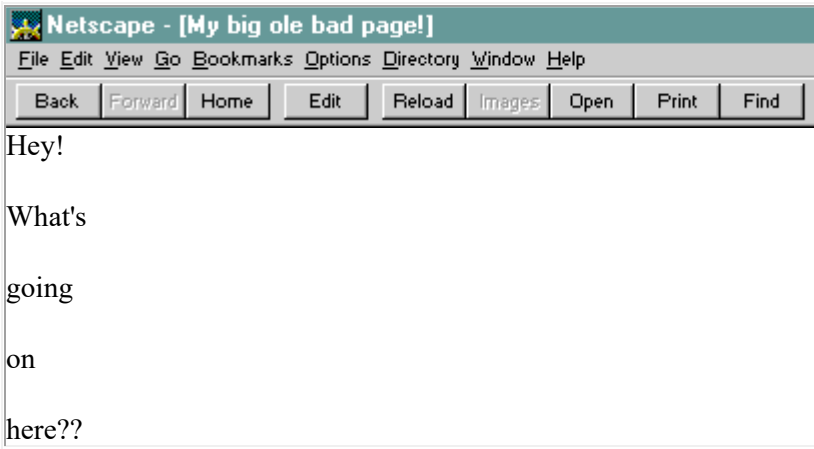
The browser doesn't recognize formatting. Unless you tell it otherwise, it just displays the characters in a steady stream. If you want to start a new line you have to use a line break.

```
<BODY BGCOLOR="#FFFFFF">
Hey!<BR>
What's<BR>
going<BR>
on<BR>
here??
</BODY>
```



 simply says- start a new line. Simililar to
 is <P>. It does exactly the same thing but it breaks, then skips a line.

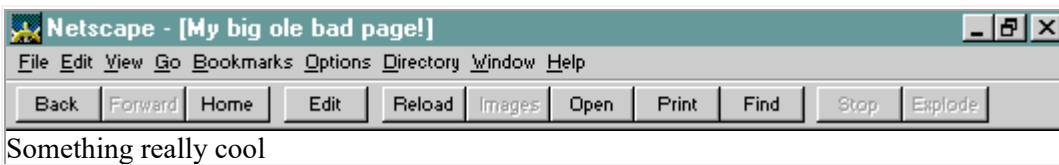
```
<BODY BGCOLOR="#FFFFFF">
Hey!<P>
What 's<P>
going<P>
on<P>
here??
</BODY>
```



These are examples of standalone tags. No closing tag required. Another thing about these line break tags... you can't use them more than once. In other words, specifying `<P><P><P>` won't give you 3 empty lines, it will just give you 1. How can you add several empty lines? I'll tell you in a minute.

Look at this first...

```
<BODY BGCOLOR="#FFFFFF">
Something          really          cool
</BODY>
```

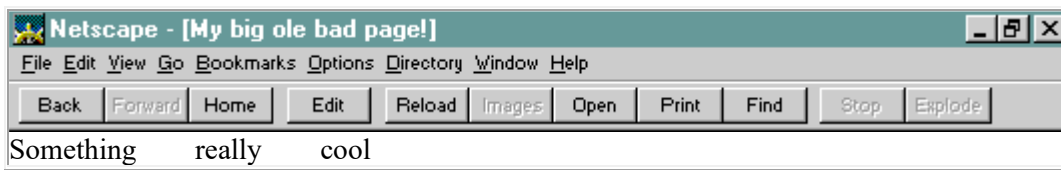


The browser won't recognize more than 1 space. I know at first this might all seem pretty stupid for it to be this way, but really, it's better to have it this way. It gives you absolute control over the document's appearance.

There is a nifty little code that means "space" to the browser -> ` `

Try this...

[illegible]



The & means we are beginning a special character the ; means ending a special character and the letters in between are sort of an abbreviation for what it's for. There are six of these special characters. (Note- these should always be lower case)

- (non-breaking space)
- < (< less-than symbol)
- > (> greater-than symbol)
- & (& ampersand)
- " (" quotation mark)
- ­ (soft hyphen)

You don't need to use them *all* the time, just when typing the real character would confuse the browser. How do you know when that is? No hard and fast 'rule' that I can think of. It will just come with a little practice and a few screw-ups.

By the way, some thoughts on mistakes & screw-ups. There are those that are of the opinion that mistakes are bad. They are afraid to try anything new for fear of messing it up. Making the same mistake over and over might be a little dumb, but especially while you are learning, don't be afraid to screw everything all up. Mistakes are our friends :)

If you are not screwing something up then you are not learning anything and probably not doing anything. Remember, messing things up is a perfectly acceptable by-product of learning!

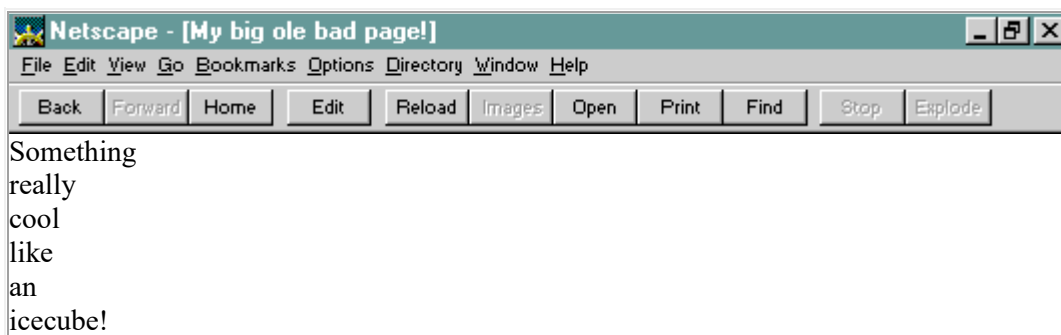
 [More](#)

OK, enough babbling. There are [other special characters](#) too. You'll probably hardly ever use them but I want you to know they are there.

Let's go over the last couple points real quick because if you're at all like me, it will get confusing. The browser will display your text in a steady stream unless you tell it otherwise with line breaks, etc. It will reduce any empty areas to 1 space. If you want more spaces, you must use the space code (). Here's a tidbit that we didn't cover... If you hit Return (or Enter) while you are typing, the browser will interpret that as a space... unless there is already a space there.

One more quick example.

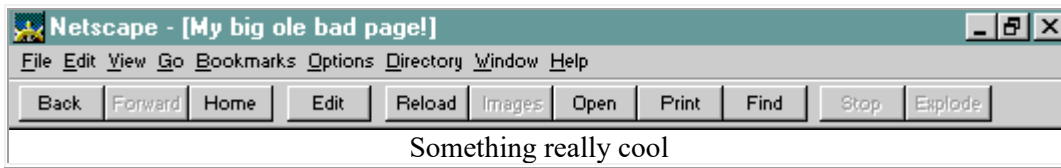
```
<BODY BGCOLOR="#FFFFFF">
Something<BR>really<BR>cool<BR>
like<BR>an<BR>icecube!
</BODY>
```



Pretty clear?? I hope so. I gave it my best shot!

Next up is a very useful little tag. It's pretty self explanatory.

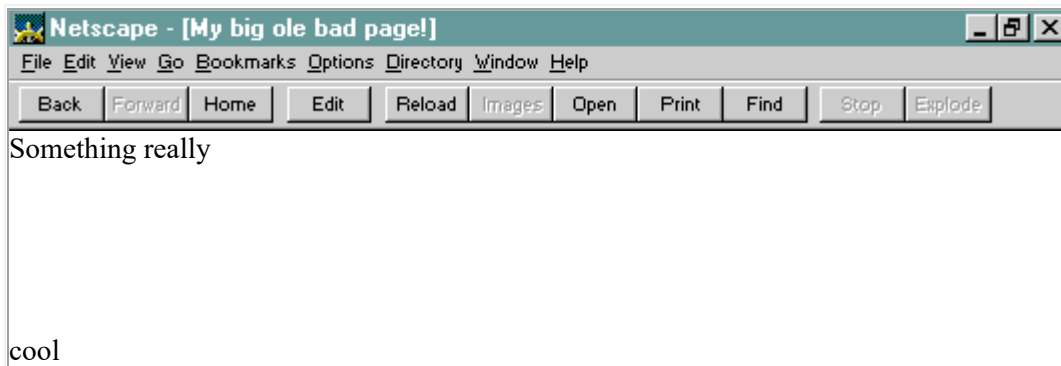
```
<BODY BGCOLOR="#FFFFFF">
<CENTER>Something really cool</CENTER>
</BODY>
```



You can center one word or your whole page. Everything between the <CENTER> tags gets centered.

I almost forgot, I was going to show you how to make multiple blank lines. It's really simple. Make an empty space with a line break for each blank line you want.

```
<BODY BGCOLOR="#FFFFFF">
Something really<BR>
&nbsp;<BR>
&nbsp;<BR>
&nbsp;<BR>
&nbsp;<BR>
&nbsp;<BR>
cool
</BODY>
```



Let's get into putting images into a web page. We're going to use this one. Once again, right click to save it off this page or copy it from the pics folder.

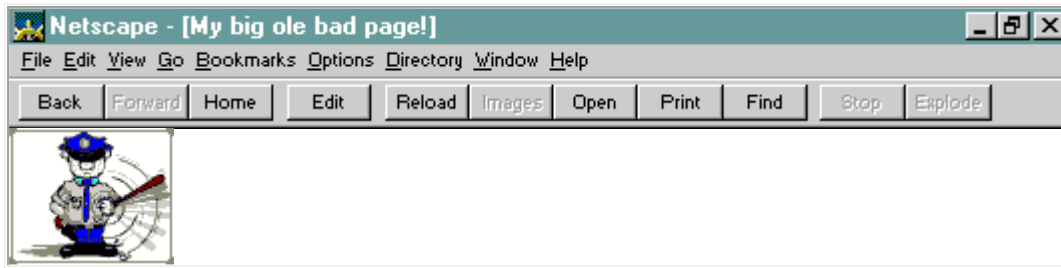
You specify an image with the (image) tag.

```
<BODY BGCOLOR="#FFFFFF">
<IMG>
</BODY>
```

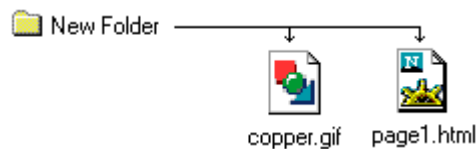


We must also specify the source and the size.

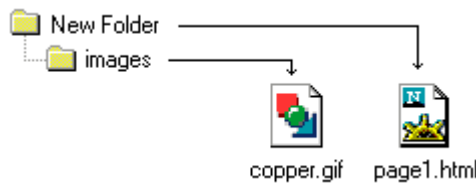
```
<BODY BGCOLOR="#FFFFFF">
<IMG SRC="copper.gif" WIDTH=82 HEIGHT=68>
</BODY>
```



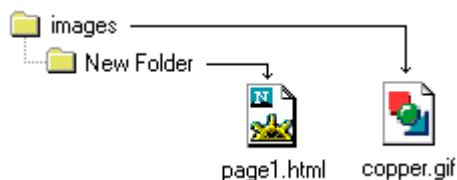
Let me make the point that not only does the source specify *what* image, it also specifies *where* is the image. The above source, "copper.gif", means that the browser will look for the image named copper.gif in the same folder (or directory) as the html document itself. Below are a few diagrams.



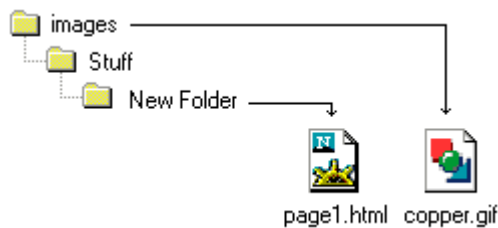
SRC="copper.gif" means that the image is in the same folder as the html document calling for it.



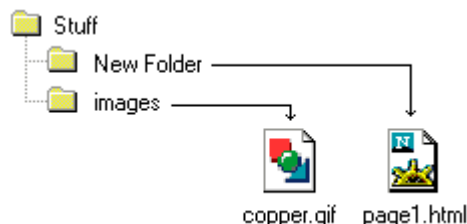
SRC="images/copper.gif" means that the image is one folder down from the html document that called for it. This can go on down as many layers as necessary.



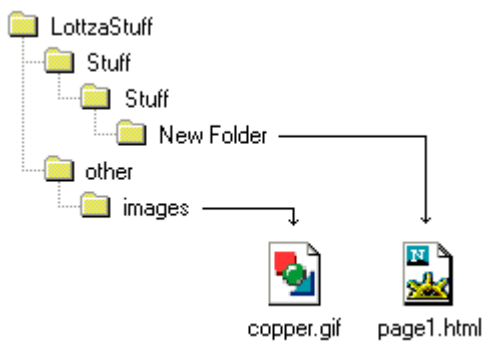
SRC="../copper.gif" means that the image is in one folder up from the html document that called for it.



SRC="../../copper.gif" means that the image is two folders up from the html document that called for it.



SRC="../images/copper.gif" means that the image is one folder up and then another folder down in the images directory.



SRC="../../other/images/copper.gif"
 I'm not even going to try and put this into words. I hope you get the drift.

There is another way that this can be done. All references to images can have as their source the complete URL. For example:

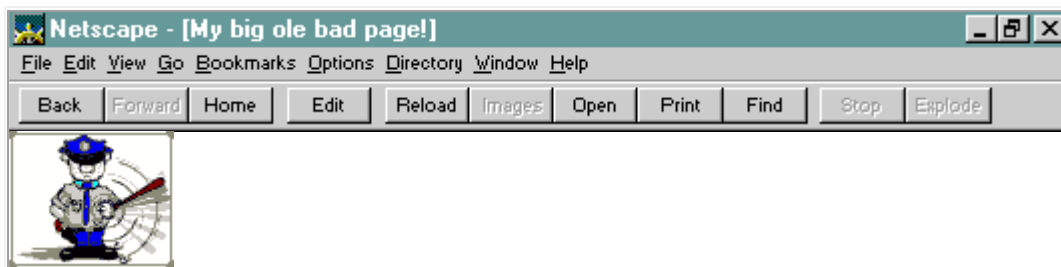
<http://www.hair.net/~squiggie/LottzaStuff/other/images/copper.gif>

Why, you ask, does it make so much more sense to use *relative* (partial) URLs as opposed to *absolute* (complete) URLs?? Because you can build your site locally and all the links will work. When your pages are done, you just upload the whole pile to your server and everything will work just fine. In addition, it is easier for the browser to get the images and your page will load faster. Is there ever a reason to use an absolute URL? Sure, if the image resides on a completely different server.

Something really neat you should know about images and their size.

Try this...

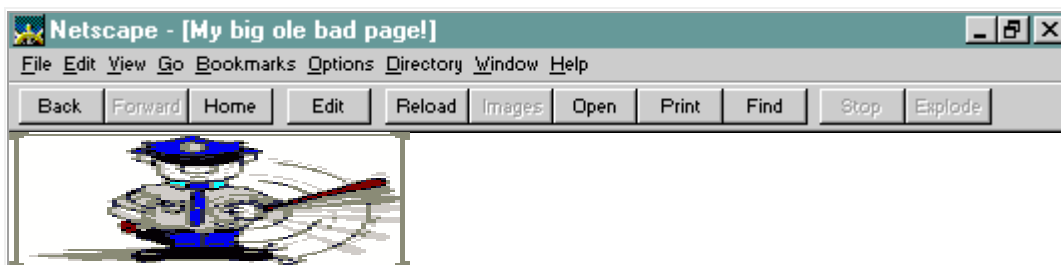
```
<BODY BGCOLOR="#FFFFFF">
<IMG SRC="copper.gif">
</BODY>
```



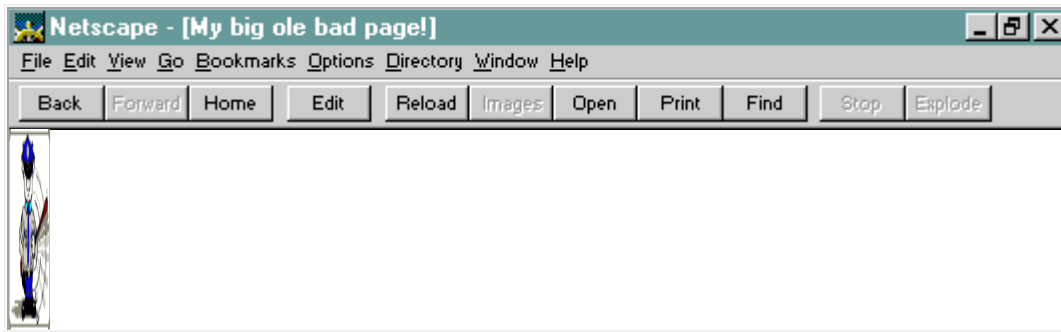
As you can see, the browser figures out how big the image is all by itself. Why bother with dimensions then? Without getting into details, it makes your page load faster because it is easier for the browser.

What's the neat part?? Check this out...

```
<BODY BGCOLOR="#FFFFFF">
<IMG SRC="copper.gif" WIDTH=200 HEIGHT=68>
</BODY>
```

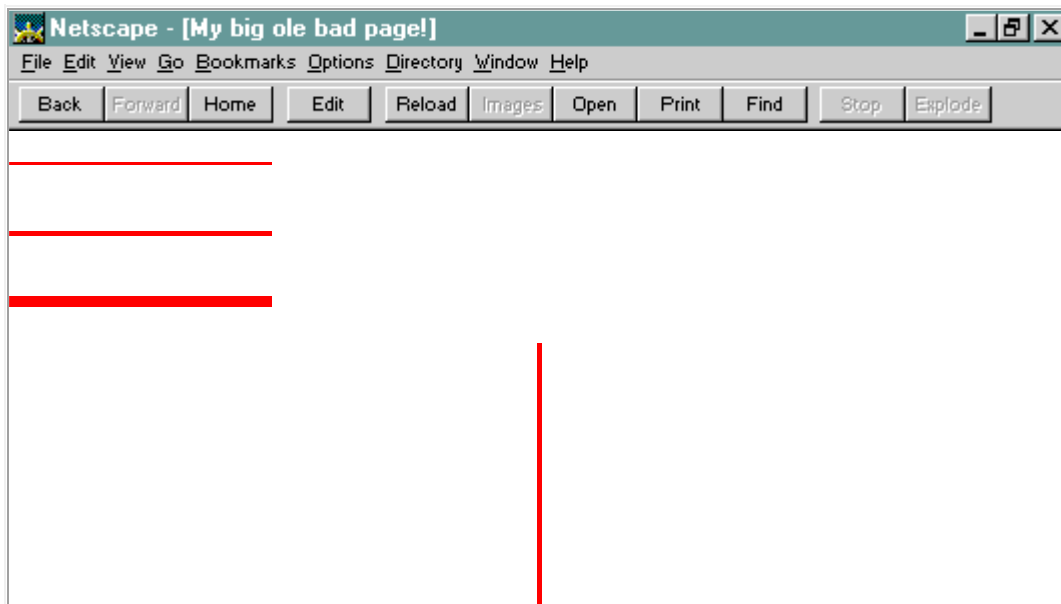



```
<BODY BGCOLOR="#FFFFFF">
<IMG SRC="copper.gif" WIDTH=20 HEIGHT=100>
</BODY>
```



You can specify whatever dimensions you want and override the proper dimensions. Still foggy on the neat part? Well, look at this little red dot-> .<- . It's a 2x2 square. Lookie what I can do with it though...

```
<BODY BGCOLOR="#FFFFFF">
<IMG SRC="red_dot.gif" WIDTH=510 HEIGHT=1><P>
<IMG SRC="red_dot.gif" WIDTH=510 HEIGHT=2><P>
<IMG SRC="red_dot.gif" WIDTH=510 HEIGHT=5><P>
<CENTER><IMG SRC="red_dot.gif" WIDTH=2 HEIGHT=200></CENTER>
</BODY>
```



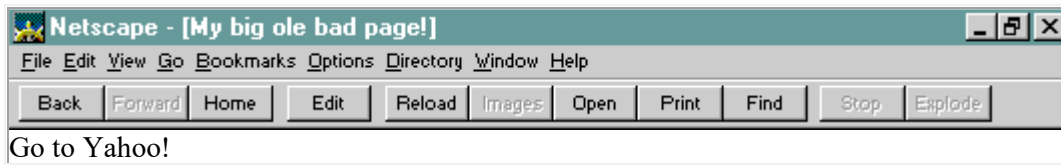
Pretty nifty huh?



Let's learn about links.

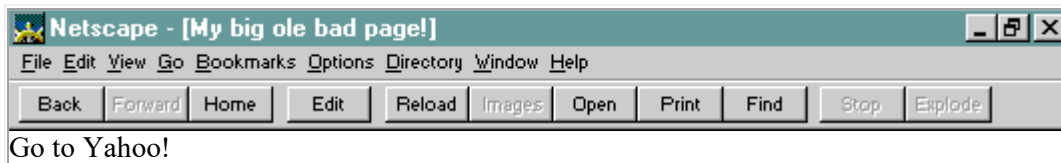
It's really very simple. We'll make a link to [Yahoo](#).
Start with this...

```
<BODY BGCOLOR="#FFFFFF">
Go to Yahoo!
</BODY>
```



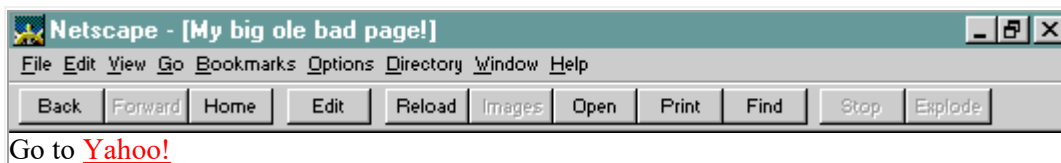
Then add a pair of anchor tags.

```
<BODY BGCOLOR="#FFFFFF">
Go to <A>Yahoo!</A>
</BODY>
```



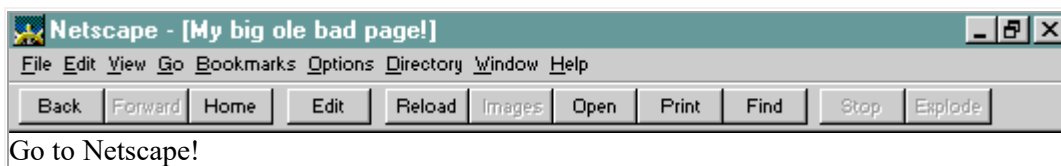
Add the URL and you're done! URL Stands for Universal Resource Locator. That's a big fancy phrase that the computer people came up with. They tend to do that alot. A URL is just an address.

```
<BODY BGCOLOR="#FFFFFF">
Go to <A HREF="http://www.yahoo.com/">Yahoo!</A>
</BODY>
```

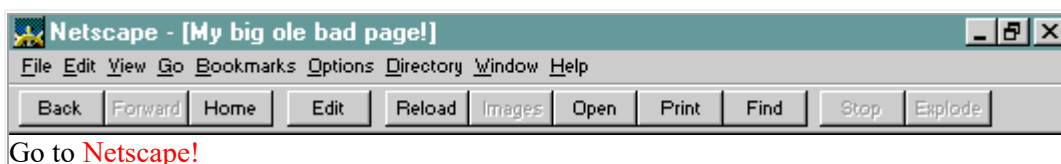


Let's do one more.

```
<BODY BGCOLOR="#FFFFFF">
Go to Netscape!
</BODY>
```



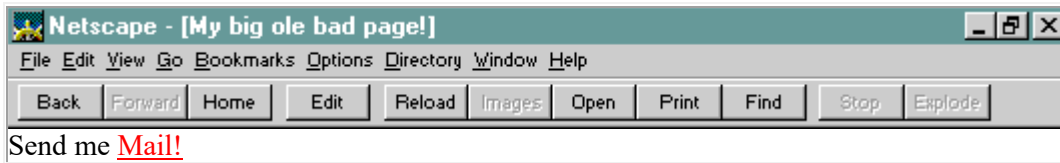
```
<BODY BGCOLOR="#FFFFFF">
Go to <A HREF="http://home.netscape.com/">Netscape!</A>
</BODY>
```





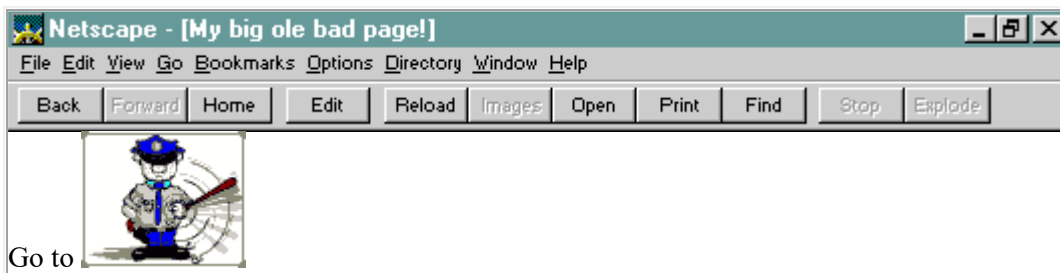
An email link works the same way. We just use an email address instead of a page address.

```
<BODY BGCOLOR="#FFFFFF">  
Send me <A HREF="mailto:forrest@bubbagump.com">Mail!</A>  
</BODY>
```



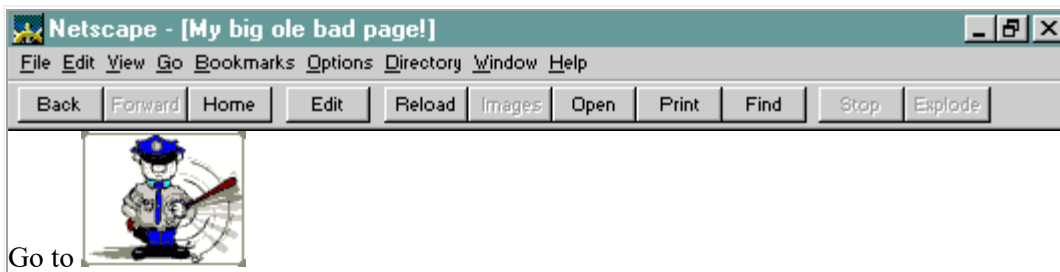
We can make an image a link if we want. Using the 'Go to Netscape!' example above we simply substitute an tag for the word Netscape!

```
<BODY BGCOLOR="#FFFFFF">  
Go to <A HREF="http://home.netscape.com/"><IMG SRC="copper.gif" WIDTH=82  
HEIGHT=68></A>  
</BODY>
```



A frequent question I get is how do you get rid of that annoying blue border around the image when you make it a link? Simple...

```
<BODY BGCOLOR="#FFFFFF">  
Go to <A HREF="http://home.netscape.com/"><IMG SRC="copper.gif" WIDTH=82  
HEIGHT=68 BORDER=0></A>  
</BODY>
```



One more note about links... It is perfectly acceptable to link to someones page(s) without asking. Links are what makes the Web the *Web*.

When you include images on your page, remember that they contain a lot of data and therefore can be slow to load. One way to reduce the size (by size I mean Kb) is to reduce the dimensions. Cutting the length and width in half results in an image only 1/4 the size in Kb. Also you can reduce the number of colors. Here are a few examples:



Dimensions 310 x 304
of colors- 238
Size - 69 Kb



Dimensions 207 x 203
of colors- 238
Size- 34 Kb

Dimensions 207 x 203
of colors- 48
Size - 19 Kb

All image editing was done with [Paint Shop Pro](#).

Now granted, the first one is a larger, nicer and better quality picture, but think of it this way... if your page takes forever to load your visitors will probably leave and they won't see *anything*.

Gif Wizard is a very helpful tool from [Raspberry Hill Publishing](#) that will help you reduce the size of your images. (Note that the GIF Wizard is not a part of this tutorial. It is a program that resides on a computer at Raspberry Hill Publishing. This is simply an interface. Also you must be online to use it.)

Step 1

One way or another they need to get your image. You have 2 options...

Option A: You can type in the URL of an image that is posted to a web server (the URL of the *image*... not the URL of the page that it's on).

Netscape users right click on an image in a web page and choose Copy Image Location. This copies the image's URL to the clipboard. From there you just paste it into the window

http://

- OR -

Option B: You can send them the image straight off your hard drive.

Definitely the "cool" way to do it. (Not supported by MS Internet Explorer)

Step 2
A & B
both are
optional

A) You can pick the appropriate background color. It is preset for white. "ND" colors are non-dithering. If not sure what to do here, just do nothing.

Red Green Blue

B) Your image can be resized if you'd like. Enter the new image size expressed either in pixels or as a percentage. Leave blank for no dimension changes.

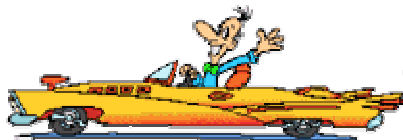
Width= Height=

Step 3

Send image to the GIF Wizard!

You will get back several versions of your image in varying degrees of "shrunkeness".

Another option that can be used if you want to offer alot of imagery is to provide *thumbnails* that link to larger versions of the image. Let's suppose that I want to offer three very nice pictures of let's say a few of my cars (I wish).



Click on the thumbnail to see a larger image

The first thing you need to do is fire up your image editor and make smaller versions of your images. It would also help to reduce the color depth. This bit about making a smaller version is very important. I have seen many instances of someone trying to make a thumbnail by simply reducing its dimensions in the tag. All this does is load the full image into a smaller space. What you need to do is create a smaller *copy* of the image and use that as a link to the big image.

Let's do one. I think we'll use the Corvette (that's the blue one if there's anyone out there that for some silly reason doesn't know). Put the full sized image and the small image in your working folder. Once again, Netscape users right click & save as.

Start with your tag.

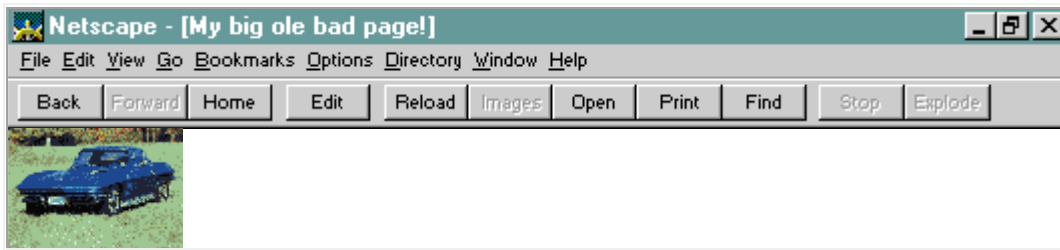
```
<BODY BGCOLOR="#FFFFFF">
<IMG SRC="car1a.gif" WIDTH=87 HEIGHT=60>
</BODY>
```





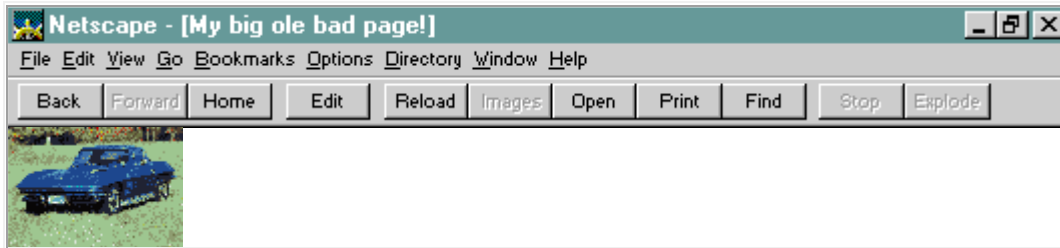
Add your <A> tags.

```
<BODY BGCOLOR="#FFFFFF">
<A><IMG SRC="carla.gif" WIDTH=87 HEIGHT=60></A>
</BODY>
```



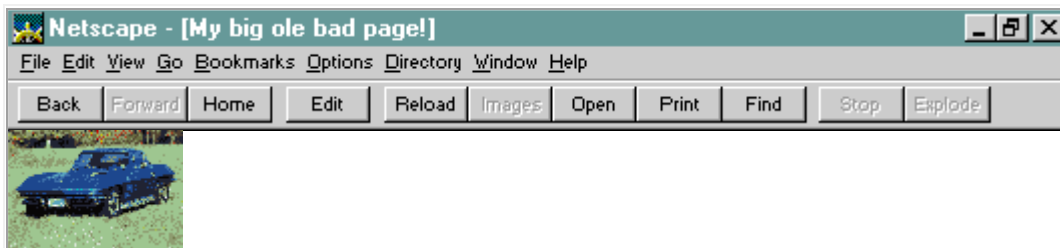
And add the URL and bingo, you're done!

```
<BODY BGCOLOR="#FFFFFF">
<A HREF="car1.jpg"><IMG SRC="carla.gif" WIDTH=87 HEIGHT=60></A>
</BODY>
```



You can, if you want, get rid of the blue border. Although, you may wish to keep it so that your viewers know that it is a link. It's up to you.

```
<BODY BGCOLOR="#FFFFFF">
<A HREF="car1.jpg"><IMG SRC="carla.gif" WIDTH=87 HEIGHT=60 BORDER=0></A>
</BODY>
```



Another way to link is to link not just to a page, but to a specific *part* of a page. Click [here](#) to be magically transported to where we first talked about URLs. Since it would be kind of hard to have you build an example, I'll just do the best I can to explain how it's done.

First start with the spot you want people to be transported *to*. Pick a word and wrap it in the <A> tags.

```
<A>Add</A> the URL and you're done!
```

Next give that spot a NAME.

```
<A NAME="upabit">Add</A> the URL and you're done!
```

What you have done is marked that spot. Now it can be referenced.

Now start building the link.

Click

```
<A>here</A>
```

 to be magically transported...

Add the document to be referenced...

Click

```
<A HREF="lesson04.html">here</A>
```

 to be magically transported...

And lastly, add the anchor NAME like so...

Click

```
<A HREF="lesson04.html#upabit">here</A>
```

 to be magically transported...

And that's all there is to it! Not exactly brain surgery is it??

We've covered alot so far. Text & font manipulation, images, links. As far as the basics go, there's not a whole heck of alot more.

I think I'm going to take this time to tell you about screen resolution. The screen I work on is 640 pixels by 480 pixels. Many use 800x600 and a few use 1024x768. I'm sure that there are even a few others out there. What does this have to do with anything? It has alot to do with how your pages will look to them. Here are a couple screen shots of my current homepage at different resolutions.



640×480



800×600



1024×768

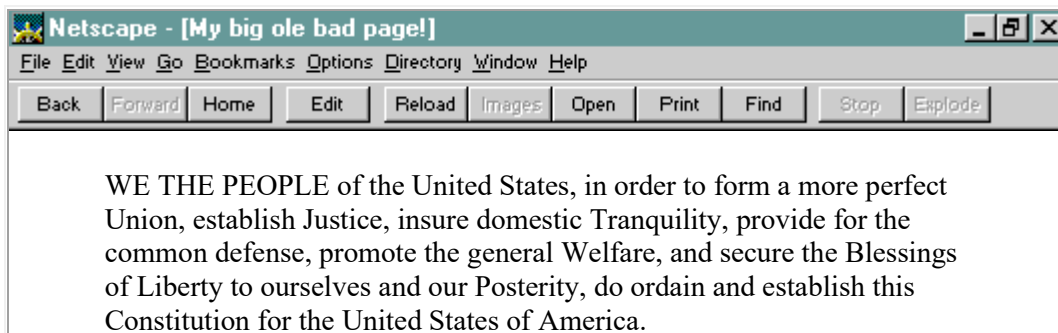
It is a very good idea to check your page at other resolutions. Your carefully crafted layout might fall apart at other resolutions. There is a handy little MS Powertoy called Quickres that lets you easily switch screen resolutions.

Now we are going to look at a couple formatting tools available to you. First one is `<BLOCKQUOTE>`. I've been using it throughout this whole tutorial. Basically it pulls your margins in from both sides. (I don't know if that's the proper terminology but if you understand what I mean I guess it's close enough).

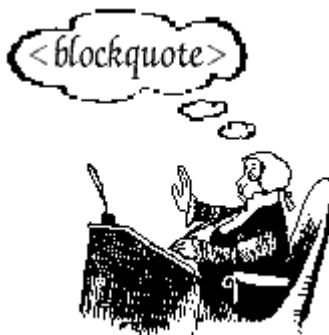
```
<BODY BGCOLOR="#FFFFFF">
<BLOCKQUOTE>
```

WE THE PEOPLE of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America.

```
</BLOCKQUOTE>
</BODY>
```



I'm sure when `<BLOCKQUOTE>` was first devised it had a loftier purpose, such as quoting profound bits of prose from authors I've never even heard of. But here in the trenches it serves a more mundane purpose... easy indenting.



Another very useful little tool is a `LIST`. There are `ORDERED` lists and `UNORDERED` lists.

This is an ordered list

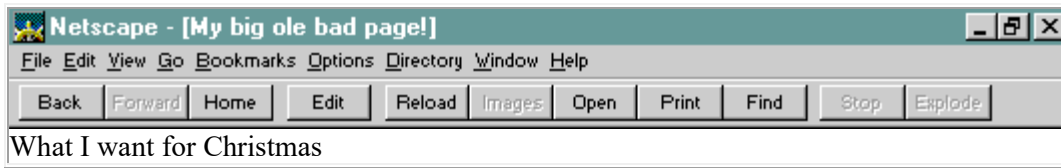
1. something big
2. something small
3. something short
4. something tall

This is an unordered list

- something red
- something blue
- something old
- something new

First, we will build an *UNORDERED* list. It's mind-numbingly simple- really. Start with this...

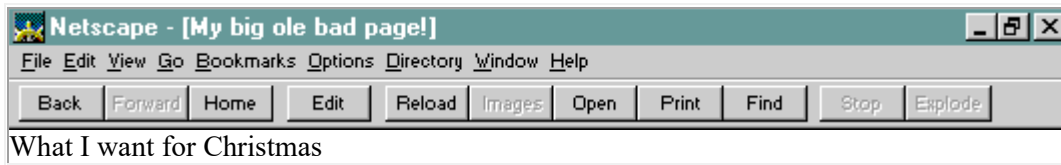
```
<BODY BGCOLOR="#FFFFFF">
What I want for Christmas
</BODY>
```



Note- technically we have not started to build the list yet. This is just a sort of heading.

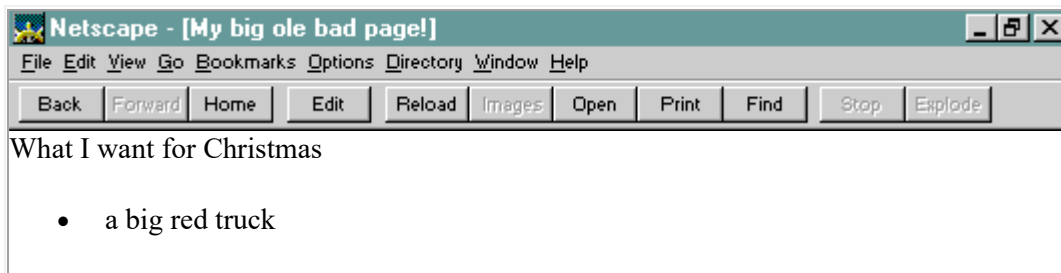
Add a pair of unordered list tags.

```
<BODY BGCOLOR="#FFFFFF">
What I want for Christmas
<UL>
</UL>
</BODY>
```



Add a list item.

```
<BODY BGCOLOR="#FFFFFF">
What I want for Christmas
<UL>
<LI>a big red truck
</UL>
</BODY>
```



Add a few more...

```
<BODY BGCOLOR="#FFFFFF">
What I want for Christmas
<UL>
```

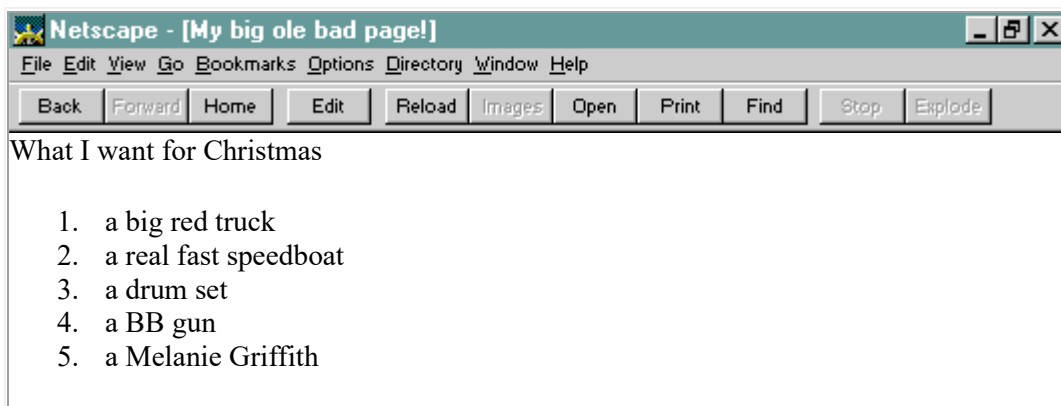
```
<LI>a big red truck
<LI>a real fast speedboat
<LI>a drum set
<LI>a BB gun
<LI>a Melanie Griffith
</UL>
</BODY>
```



Bingo! You made a list!

How to make an *ordered* list? Easy! Change the tag to .

```
<BODY BGCOLOR="#FFFFFF">
What I want for Christmas
<OL>
<LI>a big red truck
<LI>a real fast speedboat
<LI>a drum set
<LI>a BB gun
<LI>a Melanie Griffith
</OL>
</BODY>
```



Another type of list is a definition list.

aardvark

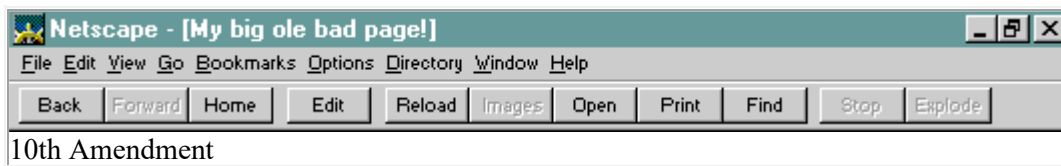
A burrowing, ant-eating African mammal. And, as everyone knows, the first word in every self respecting dictionary. Did you know there's such a thing as an *aardwolf*?

Start with this...

```
<BODY BGCOLOR="#FFFFFF">
<DL>
</DL>
</BODY>
```

Then add a definition title...

```
<BODY BGCOLOR="#FFFFFF">
<DL>
<DT>10th Amendment
</DL>
</BODY>
```



And a definition item.

```
<BODY BGCOLOR="#FFFFFF">
<DL>
<DT>10th Amendment
<DD>The powers not delegated to the United States by the Constitution, nor
prohibited by it to the States, are reserved to the States respectively,
or to the people.
</DL>
</BODY>
```



As a finishing touch I like to make the definition title bold. It's not required, but I think it looks much better that way.

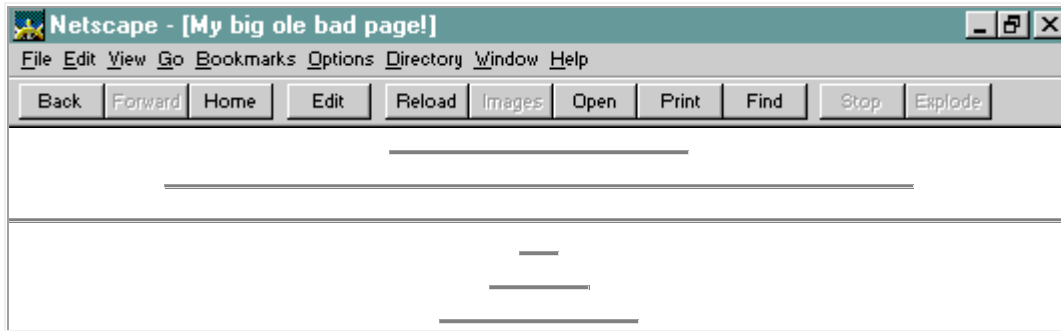
```
<BODY BGCOLOR="#FFFFFF">
<DL>
<DT><B>10th Amendment</B>
<DD>The powers not delegated to the United States by the Constitution, nor
prohibited by it to the States, are reserved to the States respectively,
or to the people.
</DL>
</BODY>
```

Another little widget that you will find yourself using alot is the Horizontal Rule.

```
<BODY BGCOLOR="#FFFFFF">
<HR>
</BODY>
```

We have a few options available to us...

```
<BODY BGCOLOR="#FFFFFF">
<HR WIDTH=20%>
<HR WIDTH=50%>
<HR WIDTH=100%>
<HR WIDTH=20>
<HR WIDTH=50>
<HR WIDTH=100>
</BODY>
```



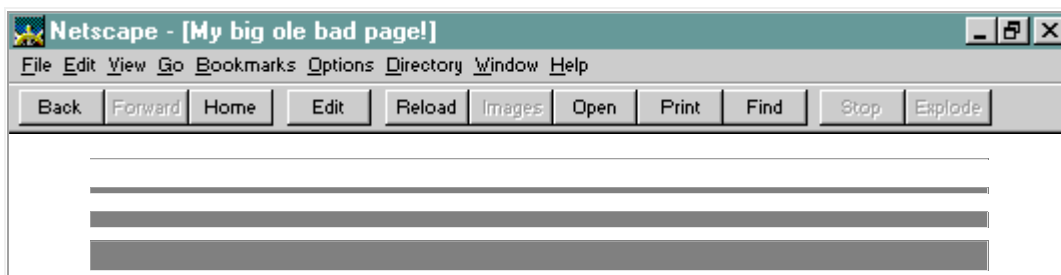
This one is pretty self explanatory.

```
<BODY BGCOLOR="#FFFFFF">
<HR WIDTH=60% ALIGN=LEFT>
<HR WIDTH=60% ALIGN=RIGHT>
<HR WIDTH=60% ALIGN=CENTER>
</BODY>
```



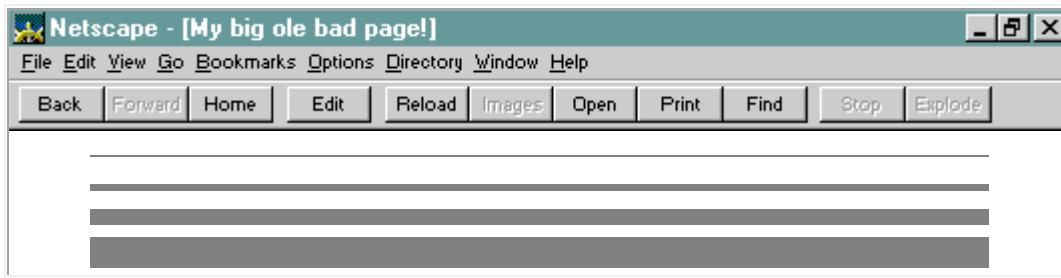
We can control the thickness...

```
<BODY BGCOLOR="#FFFFFF">
<HR WIDTH=60% SIZE=1>
<HR WIDTH=60% SIZE=3>
<HR WIDTH=60% SIZE=8>
<HR WIDTH=60% SIZE=15>
</BODY>
```



And we can make it a solid line.

```
<BODY BGCOLOR="#FFFFFF">
<HR WIDTH=60% SIZE=1 NOSHADE>
<HR WIDTH=60% SIZE=3 NOSHADE>
<HR WIDTH=60% SIZE=8 NOSHADE>
<HR WIDTH=60% SIZE=15 NOSHADE>
</BODY>
```

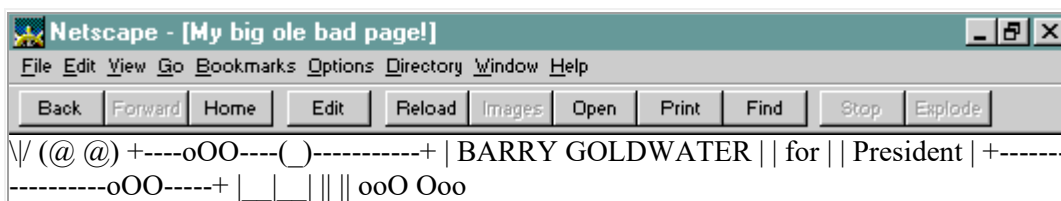


Remember when I said that the browser doesn't understand formatting, it just displays text in a steady stream?
Kind of like this...

```
<BODY BGCOLOR="#FFFFFF">

      \ | /
      (@ @)
+----oOO----( )-----+
|      BARRY GOLDWATER  |
|          for          |
|      President       |
+-----oOO-----+
      |  |  |  |
      ||  ||  ||
      ooO  Ooo

</BODY>
```

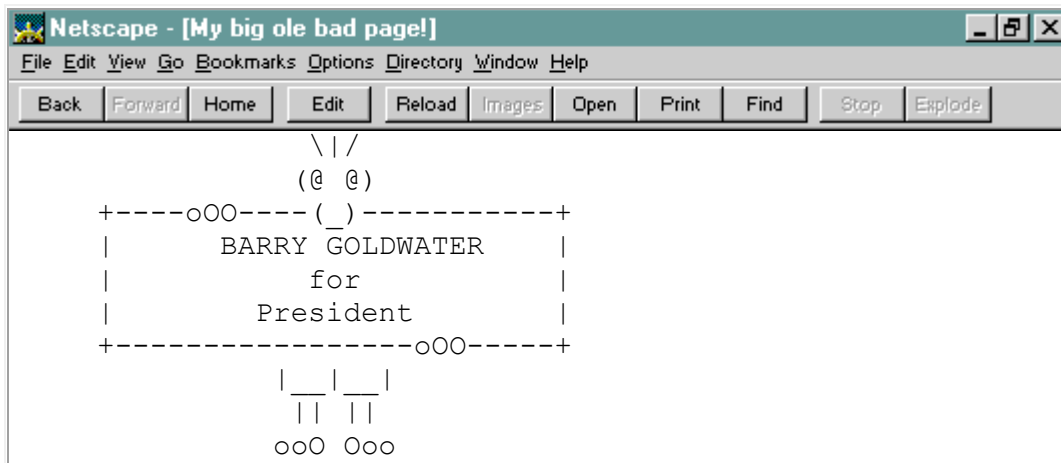


Well, with the <PRE> (preformat) tag, we can put a stop to that and have things displayed the way we type them.

```
<BODY BGCOLOR="#FFFFFF">
<PRE>

      \ | /
      (@ @)
+----oOO----( )-----+
|      BARRY GOLDWATER  |
|          for          |
|      President       |
+-----oOO-----+
      |  |  |  |
      ||  ||  ||
      ooO  Ooo
```

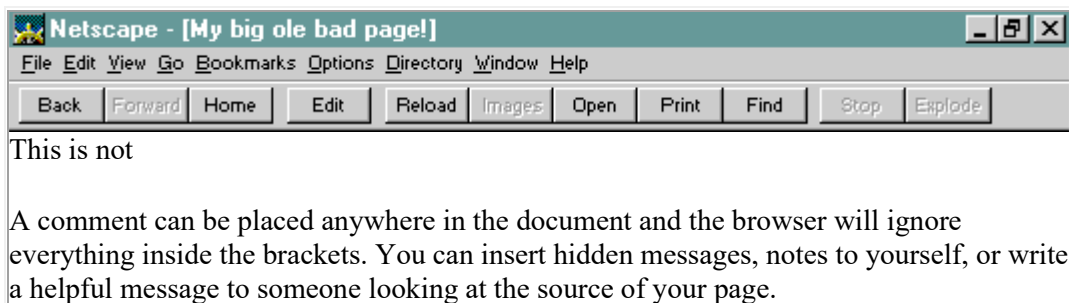
```
</PRE>
</BODY>
```



* Notice that a monospaced font is used.

The last tag we're going to discuss is a comment.

```
<BODY BGCOLOR="#FFFFFF">
<!--This is a comment-->
This is not<P>>
A comment can be placed anywhere in the document and the browser will
ignore everything inside the brackets. You can insert hidden messages, <!--
-Hi Mom!--> notes to yourself, <!--Pick up milk--> or write a helpful
message to someone looking at the source of your page.<!--Copy anything
off me and you're dead meat!-->
</BODY>
```



Just to be absolutely clear, the comment must start with <!-- and end with -->

You can even put other html tags in a comment and they will be ignored. The browser will just keep ignoring everything until it sees a -->

Well, that about does it! You now know all the basic tags you'll need to create a web page! Aren't there more tags? Oh yeah. Lots more. But rest assured you've got about 70% of the weaponry you'll need to make a web page. After you tinker with these main tags awhile, you may want to look at some of my other slightly more advanced tutorials: **Table Tutor**, **Form Tutor** and **Frames Tutor**. They will give you about another 27% of what you should know. What about that last 3% you ask?? Most of that is either tags you will probably never have occasion to use. Or, tags that for all intents and purposes duplicate other tags -

that is, perform the same function. Or, tags that have a very limited and specialized use and you can add them to your bag of tricks later.

I mentioned at the beginning of this tutorial some of my thoughts on HTML editors. In my opinion the best editors to use are not the big clunky wizard types. You know, the ones that say you don't need to know a lick of HTML to author a page. The only problem is, a large number of the pages I see that have been put together with these programs *look* as if the author didn't know a lick of HTML.

Most experienced authors use *text-based* editors. What these editors do is pop in a set of tags for you with a simple click. They make it so you don't have to manually type in color codes, LEFT, RIGHT, CENTER or whatever. You have absolute control over your page design. I once heard the phrase "text editor on steroids" to describe one. That's about what they are. A couple good ones that come to mind are listed here. What they all have in common is that you *must* know some HTML. They will make your coding easier. They don't attempt to do it for you. What do I use? Lately I've been using Super NoteTab almost exclusively.




Once you have made your pages you will need to upload them to a server. The server is a computer that is running a server program that doles out documents to whoever requests them. If you are hooked up to the internet through your company or school computer, ask your system administrator how to go about getting space on your server for your pages. If you have your own Internet Service Provider ask them how to go about it. If you are using AOL or Compuserve, etc, they may have their own ways to go about posting your pages. The only definite piece of advice I can give, is that for transferring your files, the best program to use is Cute FTP. This gem of a program makes transferring files between computers as easy as moving them around on your own computer.






Here are some excellent materials that you may want to have a look at.



INDEX & Quick Reference

Text related items









-  [Make text **bold**](#)
xxxxx
-  [Make *italics*](#)
<I>xxxxx</I>
-  [Underlining](#)
<U>xxxxx</U>
-  [Monospaced font](#)
<TT>xxxxx</TT>
-  [Change font size \(sizes:1-7\)](#)
xxxxx
-  [Change the font name \(face\)](#)
xxxxx
-  [Change the font color](#)
 xxxxx
-  [The rollercoaster!](#)
-  [The Handy Dandy Font Viewer](#)
-  [Color Picker](#)


-  [Background colors](#)
<BODY BGCOLOR="#xxxxxx">
-  [Change text, link, etc color for whole page](#) <BODY
BGCOLOR="#xxxxxx" TEXT="#xxxxxx"
LINK="#xxxxxx" VLINK="#xxxxxx"
ALINK="#xxxxxx">
-  [Line breaks](#)


-  [Paragraph break](#) <P>
-  [Space code \(special character for a space\)](#)
-  [Six main special characters](#)
-  [A whole bucketful of special characters](#)
-  [Multiple blank lines](#)

-  [Tags in combination-
Overlapping vs Nested tags](#)
-  [More about
Overlapping vs Nested tags](#)



Images

-  [Background images](#)
<BODY BACKGROUND="xxxxx.gif">
(or .jpg)
-  [Save an image off of a page](#)
-  [Inserting images](#)
<IMG SRC="xxxxx.gif" WIDTH=123
HEIGHT=456>
-  [Image sources \(SRC\) explained in detail](#)
-  [Get rid of the blue border around an image link](#)
 <IMG
SRC="xxxxx.gif" WIDTH=123 HEIGHT=456
BORDER=0>
-  [About image sizes](#)
-  [Using thumbnails](#)
-  [Make lines out of a 2x2 dot](#)


 Making an image a link
<IMG
SRC="xxxxx.gif" WIDTH=123
HEIGHT=456>


 The incredible Gif Wizard!

 About Netscape's 216 **colors**


 Netscape's 216 color chart 


Links


 Add a link abcdefg


 Add an email link

abcdefg

 Making an image a link
<IMG
SRC="xxxxx.gif" WIDTH=123
HEIGHT=456>


 Get rid of the blue border around an image link<IMG
SRC="xxxxx.gif" WIDTH=123 HEIGHT=456
BORDER=0>

 Using thumbnail images

 Link to a particular *section* of a page
abcdefg


Formatting tools


 Centering stuff <CENTER>xxxxx</CENTER>


 Line breaks


 Paragraph break <P>

 Pull in your margins with <BLOCKQUOTE>


 Ordered and Unordered lists
 (or OL)
List item


 Definition list
<DL>
<DT>Definition title
<DD>Definition item
</DL>


 Horizontal Rule
<HR ALIGN=left|right|center
WIDTH=80% SIZE=3 NOSHADE>

 The <PRE>formatting tag

Miscellaneous

 Skeleton HTML document- minimum tags required
<HTML>
<HEAD>
<TITLE>My big ole bad page!</TITLE>
</HEAD>
<BODY>

 Space code (special character for a space)

 Six main special characters

</BODY>
</HTML>



Background colors
<BODY BGCOLOR="#xxxxxx">



Document Title <TITLE>My Big ole bad
page!</TITLE>



Background images
BACKGROUND="xxxxx.gif" (or .jpg)



Tags in combination-
Overlapping vs Nested tags




More about
Overlapping vs Nested tags



About Netscape's 216 colors



Netscape's 216 color chart 



A whole bucketful of special characters



Centering stuff <CENTER>xxxxx</CENTER>



Multiple blank lines



Absolute vs Relative URLs



About screen resolution



Comment tag <!--This is a comment-->



Save document as an HTML file



Save an image off of a page



About files & file extensions



Color Picker

That's all folks !!!!



Knowing About Frames

Hello. My name is IT Kid and I'm going to give you a few simple lessons on how to make frames for your web documents. The intelligent use of frames can give your pages a cleaner look and make them easier to navigate. Note the operative word is *intelligent*. Nuff said?

Dividing a page into frames is actually quite simple. The basic concept goes like this: Each frame is a regular, complete html document. If you wanted to divide your page into 2 side by side frames, then you would put one complete html document in the left frame and another complete html document in the right frame. In addition you need to write a *third* html document. This *MASTER PAGE* (I'm coining a new phrase here folks) contains the <FRAME> tags that specify what goes where. As a matter of fact, that's its only function.

There are only two main frame tags to contend with: <FRAMESET> and <FRAME>. The easiest way to explain them is to have you dive right in and start making some frames. I guarantee that in 90 minutes you'll be a feel-no-pain framing machine!

Now's a good time to stress that if you want to learn how to make quality html documents, then you would be well served to take the time to teach yourself the tags. If you rely on the so-called "frame wizards" in the "easy as pie html editors" out there, you will have greatly limited flexibility, and the end result may not be what you are trying to achieve. In my opinion the best html editors to use are text based editors. A few good ones that come to mind are listed here. These editors will make your html coding easier. They don't attempt to do it for you.

Frames Tutor- Lesson 1

First things first. For this tutorial we will need a few html documents. Fire up Notepad and copy the following to get you started.

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>

</BODY>
</HTML>
```

We will give each document a name. In Table Tutor and Form Tutor we used guy's names. I think its time to get some young ladies in here.

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>
Lisa
</BODY>
</HTML>
```

Create a new folder somewhere and save this as `lisa.html` (Win3.x users save it as `lisa.htm`).

Note- I will be refering to documents as `xxxx.html`. Users of Windows 3.x will have to use the file extension `.htm` when working with documents locally. Many FTP programs allow you to change the filename when uploading if you need to.

Now make another html document.

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>
Terri
</BODY>
</HTML>
```

Save this in the same folder as `terri.html`.

Now do the same for Kim, Tina, Shannon, and Beth. Save them just like the others. You should now have a folder that contains 6 complete standalone html documents.

Ok, now the fun stuff... making your master page. Start with this.

```
<HTML>
<HEAD>
<TITLE>My Framz Page- The Master Page</TITLE>
</HEAD>

<BODY>
</BODY>
</HTML>
```

Remove the <BODY> tags. The master page doesn't use them...

```
<HTML>
<HEAD>
<TITLE>My Framz Page- The Master Page</TITLE>
</HEAD>

</HTML>
```

...it uses <FRAMESET> tags instead.

```
<HTML>
<HEAD>
<TITLE>My Framz Page- The Master Page</TITLE>
</HEAD>

<FRAMESET>
</FRAMESET>
</HTML>
```

To keep things a little cleaner, I'm going to stop writing the <HTML>, <HEAD> and <TITLE> tags. Needless to say, keep them in your document.

```
<FRAMESET>
</FRAMESET>
```

Now's a good time to save it. Save it in your folder (with all the girls) as `index.html`. If you try to open it with your browser now it will be blank. All you have said so far is "This is my Master Page".

Now let's start defining just how things are going to look. Tell the browser to split the main window into 2 columns, each occupying 50% of the window.

```
<FRAMESET COLS="50%, 50%">
</FRAMESET>
```

This will still be blank, we have one more thing to do before our 'system is operational'.

We must tell the browser what to put in each frame.

```
<FRAMESET COLS="50%, 50%">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#). You are now the proud parent of a fully functional *Framed* page!

One last thing before we go on. Note that <FRAMESET> is a *container* tag, and <FRAME> is not. For those that don't know what that means, a container tag has an opening <TAG> and a closing </TAG>.

Frames Tutor- Lesson 2

The <FRAMESET> tag does all the dividing. That's all it does... divide up windows. It specifies a few things regarding *how* to divide them up, but remember, whenever you want to do some dividing- use <FRAMESET>.

Can we divide it into more than 2 pieces? Yes, just make sure that you specify a page for each section or the browser's gonna get confused.

```
<FRAMESET COLS="20%, 20%, 20%, 20%, 20%">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
  <FRAME SRC="kim.html">
  <FRAME SRC="tina.html">
  <FRAME SRC="shannon.html">
</FRAMESET>
```

[VIEW IT](#)

It's pretty obvious that we can make the frames all different sizes. Just make sure your arithmetic is correct or the browser will come up with its own interpretation.

```
<FRAMESET COLS="10%, 20%, 30%, 15%, 25%">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
  <FRAME SRC="kim.html">
  <FRAME SRC="tina.html">
  <FRAME SRC="shannon.html">
</FRAMESET>
```

[VIEW IT](#)

If we specify dividing into ROWS instead of COLS we get something else entirely.

```
<FRAMESET ROWS="10%, 20%, 30%, 15%, 25%">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
  <FRAME SRC="kim.html">
  <FRAME SRC="tina.html">
  <FRAME SRC="shannon.html">
</FRAMESET>
```

[VIEW IT](#)

Let's go back down to 2 frames, divided equally into columns.

```
<FRAMESET COLS="50%, 50%">
  <FRAME SRC="lisa.html">
```

```
<FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

We can specify 50 pixels instead of 50%. And, we can use * instead of a number. The * means whatever is left over.

```
<FRAMESET COLS="50, *>
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

Here is an important point.. Let's suppose that you make a frame 100 pixels wide on the left, and 100 pixels wide on the right. Since you are running an 800x600 screen you make the 3rd and middle area 600 pixels wide. Everything looks just dandy... to you. My screen is set at 640x480. Those 100 pixel wide frames you put in your page are now 80 pixels wide on my screen. If you use any absolute dimensions in your <FRAMESET> tags ALWAYS have at least one * as an elastic frame. That way everything will look good to everyone and there will be peace across the land once again.

This is the cause of a common problem with frames that I want to take the time to draw extra attention to. A popular frame layout is a narrow directory window on the left and a larger main window on the right [like so](#). Here's the problem... If you divide up the windows using percentages (such as <FRAMESET COLS="15%, 85%">) everything will probably look just dandy [to you](#). But... to someone with a different screen resolution it might look [like this](#).

The reason this happens is because you have specified that the left window should be 15%. 15% of *what*? 15% of whatever their horizontal screen resolution is. That means that that left window might look a little different to different people. How to fix you say? Glad you asked... Just use an absolute dimension for the left window and make the right window elastic (such as <FRAMESET COLS="120, *>). Bingo. Problem solved.

It's good idea to make that absolute dimension just a wee bigger than it needs to be. Give the stuff in that left window a little breathing room. For example, if 100 pixels gives a good but snug fit... make the window 120 or 125 pixels.

A good and painless way to check your layout at a couple different screen resolutions is with a nifty little MS PowerToy called [Quickres](#).

We can have more than one leftover frame and specify a size relationship between them.

```
<FRAMESET COLS="50, *, 2*">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
  <FRAME SRC="kim.html">
</FRAMESET>
```

[VIEW IT](#)

Translated this says: Make 3 frames. Make the first 50 pixels wide. The rest divide between frames 2 and 3 but make frame 3 twice as big as frame 2. Put Lisa in the first frame, Terri in the second and Kim in the third.

It is important to note that everything is done in order. The first <FRAME> is displayed according to the first size attribute in the <FRAMESET> tag (50/lisa) and the second with the second (*/terri) and the third with the third (2*/kim). I know that this may be mind-numbingly obvious, but it is very important and I wanted to draw extra attention to it.

What if we want to divide kim in half horizontally? Remember that I said that if you want to do any dividing you must use the <FRAMESET> tag. First we must replace Kim with a <FRAMESET> tag pair.

```
<FRAMESET COLS="50,*,2*">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
  <FRAMESET>
  </FRAMESET>
</FRAMESET>
```

At this point we have told the browser: Make 3 frames. Make the first 50 pixels wide. The rest divide between frames 2 and 3 but make frame 3 twice as big as frame 2. Put Lisa in the first frame, Terri in the second and the third frame we are going to divide further. Now we have to specify *how* to divide up that third frame.

We were going to divide it in half horizontally.

```
<FRAMESET COLS="50,*,2*">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
  <FRAMESET ROWS="50%,50%">
  </FRAMESET>
</FRAMESET>
```

And I think we'll put Kim back in on the top and Tina on the bottom. And that's that.

```
<FRAMESET COLS="50,*,2*">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
  <FRAMESET ROWS="50%,50%">
    <FRAME SRC="kim.html">
    <FRAME SRC="tina.html">
  </FRAMESET>
</FRAMESET>
```

[VIEW IT](#)

That was so much fun I think we'll do it again! Let's divide Lisa horizontally into 3 sections. The top window needs to be 50 pixels high. The other two can be split equally. We will put Lisa into all three. Here are all the changes in one big chunk.

```
<FRAMESET COLS="50,*,2*">
  <FRAMESET ROWS="50,*,*">
    <FRAME SRC="lisa.html">
    <FRAME SRC="lisa.html">
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
  <FRAMESET ROWS="50%,50%">
    <FRAME SRC="kim.html">
    <FRAME SRC="tina.html">
  </FRAMESET>
</FRAMESET>
```

[VIEW IT](#)

I know it doesn't look like much. Actually if a page was divided up that much it would look pretty crappy. A good rule of thumb is don't have more than 3 frames on your page. If you can avoid them altogether, that's even better.

Now's also a good time to think about the poor folks that are using non frames capable browsers. Although most, if not all of your visitors will be able to see frames, there is a hapless minority that's trying to peruse the web with outdated equipment. If you would like to help these folks out you can write a non frames version of your main page and add it thusly...

```

<FRAMESET COLS="50,*,2*">
  <FRAMESET ROWS="50,*,*">
    <FRAME SRC="lisa.html">
    <FRAME SRC="lisa.html">
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
  <FRAMESET ROWS="50%,50%">
    <FRAME SRC="kim.html">
    <FRAME SRC="tina.html">
  </FRAMESET>
</FRAMESET>

```

<NOFRAMES>

<BODY>

Put your non-frames page down here. If someone is using an old browser, it will skip everything above and come straight down here. Frames-capable browsers will ignore what is between the <NOFRAMES> tags. You may be asking a couple questions right now.

Do I have to write a whole nuther page? Probably. How many people will be missing my page if I don't do this? Very few.

Do you think I should worry about it? Nope.

Do you worry about it? Nope.

Why did you put it in this tutorial? Because it's a major feature and I wanted you to at least know about it.

Are you always this witty and charming? Of course.

</BODY>

</NOFRAMES>

Note- My personal opinion aside, there is a strong argument for using the <NOFRAMES> tag. Quite a few businesses are still using older, non-frames capable browsers. The handicapped's screen readers have a really hard time with frames. And those outside the US are often left out of framed pages for various reasons. The bottom line is, if your page is "frames only" you will be cutting off some people. If it is important to you to reach everyone, then take the time to use the <NOFRAMES> tag. -----My thanks to Matthew Miller, of the [HTML Writers Guild](#) for pointing this out.

Frames Tutor- Lesson 3

Let's go back to something a wee bit simpler.

```

<FRAMESET COLS="50%,50%">
  <FRAME SRC="lisa.html">
  <FRAME SRC="terri.html">
</FRAMESET>

```

[VIEW IT](#)

We can put pictures in the frames if we want to. In the folder named 'examples' there is a picture called world.gif. Go ahead and copy that into your working folder.

```

<FRAMESET COLS="50%,50%">
  <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162>
  <FRAME SRC="terri.html">
</FRAMESET>

```

[VIEW IT](#)

Remember it is always good practice to include HEIGHT & WIDTH attributes for any image tag.

Let's try to frame the picture nice and neat. First shrink the left window to 146 pixels wide. Since we are using an absolute dimension should make the other frame elastic.


```
<FRAMESET COLS="146,*">
  <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162>
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

Next we will divide the left frame horizontally into 2 sections. The top section to be 162 pixels high and the bottom section to be whatever is left over. We will put in world.gif and Lisa respectively.

```
<FRAMESET COLS="146,*">
  <FRAMESET ROWS="162,*">
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

As you can see, we have a bit of a problem.

The scrollbars that you see can be specified as YES, NO or AUTO. YES means the window gets scrollbars- whether they're needed or not. NO means there will be no scrollbars, even if your frame contents are as big as New Jersey... the browser will simply display as much as it can. AUTO is the default. If scrollbars are needed, they appear, if they are not needed they stay conveniently out of the way. Let's get rid of our scrollbars.

```
<FRAMESET COLS="146,*">
  <FRAMESET ROWS="162,*">
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162 SCROLLING=NO>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
</FRAMESET>
```

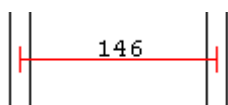
[VIEW IT](#)

We still have a problem. The image is not in the frame right. The next two attributes deal with *margins*. The browser automatically gives each frame some empty space around its contents. This is normally necessary for aesthetics. You can control the size of these margins by using MARGINWIDTH and MARGINHEIGHT. They control the left & right and top & bottom margins respectively. We will set them both to 1. (1 is the minimum)

```
<FRAMESET COLS="146,*">
  <FRAMESET ROWS="162,*">
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162 SCROLLING=NO
      MARGINWIDTH=1 MARGINHEIGHT=1>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

Unfortunately that still doesn't quite do it. The frame dimensions are measured center to center.



There is also that minimum border of 1 to deal with. Since the default border size is 6 pixels, we have to add 8 to each image dimension (half of 6, plus 1, times 2). If that's confusing, just add 8 to each image dimension to get your window size.

```
<FRAMESET COLS="154,*">
  <FRAMESET ROWS="170,*">
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162 SCROLLING=NO
      MARGINWIDTH=1 MARGINHEIGHT=1>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

This will display your image perfectly in the window, or at least pretty darn close. Have you ever seen an [image](#) in a frame that was cut off on one or more sides? If you use absolute dimensions, along with an elastic frame... And if you add 8 to the size of the image to get your frame dimensions, by golly you'll look just fine to pert near everyone. [*](#)

Frames Tutor- Lesson 4

Next we'll try a couple minor effects. Not all of them work with all browsers though. If the browser doesn't understand the attribute, it will just ignore it.

First we can change the thickness of the borders.

```
<FRAMESET COLS="154,*" BORDER=20>
  <FRAMESET ROWS="170,*">
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162 SCROLLING=NO
      MARGINWIDTH=1 MARGINHEIGHT=1>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

We can change the *color* of the borders.

```
<FRAMESET COLS="154,*" BORDER=20 BORDERCOLOR="#FF0000">
  <FRAMESET ROWS="170,*">
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162 SCROLLING=NO
      MARGINWIDTH=1 MARGINHEIGHT=1>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

We can turn off borders for individual <FRAMESET>s with FRAMEBORDER.


```
<FRAMESET COLS="154,*" BORDER=20 BORDERCOLOR="#FF0000">
  <FRAMESET ROWS="170,*" FRAMEBORDER=NO >
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162 SCROLLING=NO
      MARGINWIDTH=1 MARGINHEIGHT=1>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

We can prevent the viewer from resizing a frame. Unless you have a special circumstance, there really is no reason to use this attribute alot.

```
<FRAMESET COLS="154,*" BORDER=20 BORDERCOLOR="#FF0000">
  <FRAMESET ROWS="170,*" FRAMEBORDER=NO >
    <FRAME SRC="world.gif" WIDTH=146 HEIGHT=162 SCROLLING=NO
      MARGINWIDTH=1 MARGINHEIGHT=1>
    <FRAME SRC="lisa.html">
  </FRAMESET>
  <FRAME SRC="terri.html" NORESIZE>
</FRAMESET>
```

[VIEW IT](#)

Ok. Now that we have learned a bit about Framing  let's go back to something simple and we'll tear into *linking* between frames.

```
<FRAMESET COLS="33%,67%">
  <FRAME SRC="beth.html">
  <FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

What we're going to do is add a link from `beth.html` to `shannon.html`. So open `beth.html` with Notepad and add the following...

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>
Beth<P>
You have to visit my friend Shannon
</BODY>
</HTML>
```

Then add the link, and save it.

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>
Beth<P>
You have to visit my friend <A HREF="shannon.html">Shannon</A>
</BODY>
</HTML>
```

Now try it.

```
<FRAMESET COLS="33%,67%">
  <FRAME SRC="beth.html">
```

```
<FRAME SRC="terri.html">
</FRAMESET>
```

[VIEW IT](#)

If you click on that link you'll see Shannon's page load into Beth's window. I know, I know... you're not impressed. You want to click on a link in Beth's window and have it load into *Terri's* window. Well, that's a little more work.

What you have to do is NAME the <FRAME> in your master page. So go ahead and give that second frame a name.

```
<FRAMESET COLS="33%,67%">
  <FRAME SRC="beth.html">
  <FRAME SRC="terri.html" NAME="WINDOW-1">
</FRAMESET>
```

Let me make a comment here. That second frame is not really "Terri's". It's simply the second frame and we have specified that `terri.html` is the first thing to be loaded into it. If we had an empty <FRAME> tag the window would be... well... empty.

Note- <FRAME> NAMES must *begin* with an alpha-numeric character. All other window names will be ignored. (An exception is to begin the frame name with the underscore _ Its use is explained a little later.)

Next we have to add a little something to that link in Beth's page, so open `beth.html` again (with Notepad) and add a TARGET.

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>
Beth<P>
You have to visit my friend <A HREF="shannon.html" TARGET="WINDOW-1">Shannon</A>
</BODY>
</HTML>
```

This will cause the link to load into the window named WINDOW-1.

[VIEW IT](#)

That kind of linking is great when you want to link together pages in your site. But what if you want to link to something outside your site? Let's add another link to Beth's page.

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>
Beth<P>
You have to visit my friend <A HREF="shannon.html" TARGET="WINDOW-1">Shannon</A><P>
And of course you have to visit Joe at <A HREF="http://junior.apk.net/~jbarta/"
TARGET="WINDOW-1">Professional Web Design</A><P>
</BODY>
</HTML>
```

[VIEW IT](#)

As you can see, this causes someone else's page to load into your window

How can you make an outside link load into the full browser window? Easy, change the TARGET to `_top`.

```
<HTML>
<HEAD>
<TITLE>My Framz Page</TITLE>
</HEAD>
<BODY>
Beth<P>
You have to visit my friend <A HREF="shannon.html" TARGET="WINDOW-1">Shannon</A><P>
And of course you have to visit Joe at <A HREF="http://junior.apk.net/~jbarta/"
TARGET="_top">Professional Web Design</A><P>
</BODY>
</HTML>
```

VIEW IT

Always use this method when linking to an outside page. Believe me, your viewers will appreciate it!

`_top` is one of 4 so-called 'magic targets'. They are `_self`, `_blank`, `_parent` and `_top`. These are the only targets that can begin with something other than an alpha-numeric character. In addition, any target beginning with an underscore `_` that is *not* one of the 'magic targets' will be ignored. For what we're doing, `_top` is the only one to concern ourselves with now.

Note- It is important to specify `TARGET="_top"` rather than `TARGET="_TOP"`. Normally HTML is not case-sensitive but in this instance it is. Using `_TOP` instead of `_top` will sometimes cause the link to open in a new browser rather than the full window of the existing browser. Since it's been mentioned, that's what `TARGET="_blank"` does.... loads a link into a new browser window.

Frames Tutor- Lesson 5

I think what we'll do now is make a Framed Page from scratch.
Nothing fancy, just something like [this](#).

Before we start, let me stress... keep it simple! A site with a bunch of frames and links pointing all over the place is going to be confusing to build and even more confusing to navigate. That said, let's go to it!

First we must think about what we want the end result to be. I think a good simple design would be a banner across the top, a directory window on the left, and the main window on the right. Let's make the Master Page first.

```
<HTML>
<HEAD>
<TITLE>A Practice Page</TITLE>
</HEAD>
```

```
<FRAMESET>
</FRAMESET>
</HTML>
```

Make another new folder and save this as `index.html`. Also the image we will be using for the banner is the "MY FRAMZ PAGE" image. It is called `framz1.gif` and it is in the `framz` folder. Copy it into the working folder you just created.

Divide the screen horizontally.

```
<HTML>
<HEAD>
<TITLE>A Practice Page</TITLE>
</HEAD>
```

```
<FRAMESET ROWS="83, *">
</FRAMESET>
</HTML>
```

Note we get 83 by adding 8 to the height of the image ($75+8=83$). Also note that since we are using an absolute dimension we are including an elastic frame.

Next specify `banner.html` to go in the top frame (we will make that document in a couple minutes). Also throw in a `<FRAMESET>` tag pair because we are going to divide that bottom portion further.

```
<HTML>
<HEAD>
<TITLE>A Practice Page</TITLE>
</HEAD>

<FRAMESET ROWS="83,*">
  <FRAME SRC="banner.html">
  <FRAMESET>
  </FRAMESET>
</FRAMESET>
</HTML>
```

[VIEW IT](#)

You can run this now but you may get error messages because it's incomplete.

We will divide the bottom window into two sections. We will also specify that the left window contain `directry.html` and the right window contain `home.html`. Once again, we have not made these documents so you may still get an error message (or two) (or three).

```
<HTML>
<HEAD>
<TITLE>A Practice Page</TITLE>
</HEAD>

<FRAMESET ROWS="83,*">
  <FRAME SRC="banner.html">
  <FRAMESET COLS="20%,80%">
    <FRAME SRC="directry.html">
    <FRAME SRC="home.html">
  </FRAMESET>
</FRAMESET>
</HTML>
```

[VIEW IT](#)

Since we will have our directory on the left and pages will load into the righthand frame, we should name that frame. Its the only one that will have stuff loaded into it so its the only one we really need to name.

```
<HTML>
<HEAD>
<TITLE>A Practice Page</TITLE>
</HEAD>

<FRAMESET ROWS="83,*">
  <FRAME SRC="banner.html">
  <FRAMESET COLS="20%,80%">
    <FRAME SRC="directry.html">
    <FRAME SRC="home.html" NAME="MAIN-WINDOW">
  </FRAMESET>
</FRAMESET>
</HTML>
```

[VIEW IT](#)

OK. We are done with this one... for now.

Lets make `banner.html`. Start with the following and save it.

```
<HTML>
<HEAD>
<TITLE>Practice Page- Banner</TITLE>
</HEAD>
<BODY BGCOLOR="#0000FF">
</BODY>
</HTML>
```

Notice we have specified a background color of blue.

[VIEW banner.html](#)

[VIEW Master Page](#)

Pop in the image and <CENTER> it.

```
<HTML>
<HEAD>
<TITLE>Practice Page- Banner</TITLE>
</HEAD>
<BODY BGCOLOR="#0000FF">
<CENTER><IMG SRC="framz1.gif" WIDTH=500 HEIGHT=75></CENTER>
</BODY>
</HTML>
```

[VIEW banner.html](#)

[VIEW Master Page](#)

While I'm thinking about it, I just wanted to mention that if you look at the source of *my* documents to assist you in creating *your* documents you may find a few things that don't make sense or that may contradict what I say. That is because I have to make *my* pages jump through a few hoops so that the lessons work right when viewed. Make sense? If not, just don't look at the source of these pages unless you want to confuse yourself.

Alright... as you can see we have a problem. We have a scrollbar getting in the way and the image is not positioned in the window very well. Open your Master Page. Turn off the scroll bars and get rid of the margins.

```
<HTML>
<HEAD>
<TITLE>A Practice Page</TITLE>
</HEAD>

<FRAMESET ROWS="83,*">
  <FRAME SRC="banner.html" SCROLLING=NO MARGINWIDTH=1 MARGINHEIGHT=1>
  <FRAMESET COLS="20%,80%">
    <FRAME SRC="directry.html">
    <FRAME SRC="home.html" NAME="MAIN-WINDOW">
  </FRAMESET>
</FRAMESET>
</HTML>
```

[VIEW IT](#)

Now let's make our Directory page. Start with the following and save it as `directry.html`.

```
<HTML>
<HEAD>
<TITLE>Practice Page- Directory</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
</BODY>
</HTML>
```

Give it a heading and write in the text of all the links. We'll add the link info in a minute.

```
<HTML>
<HEAD>
<TITLE>Practice Page- Directory</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">

<H3>Directory</H3>
Home<P>
Go here<BR>
or there<P>
or visit<BR>
Yahoo<BR>
Netscape
</BODY>
</HTML>
```

Now add the link information.

```
<HTML>
<HEAD>
<TITLE>Practice Page- Directory</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">

<H3>Directory</H3>
<A HREF="home.html">Home</A><P>
<A HREF="here.html">Go here</A><BR>
<A HREF="there.html">or there</A><P>
or visit<BR>
<A HREF="http://www.yahoo.com/">Yahoo</A><BR>
<A HREF="http://home.netscape.com/">Netscape</A>
</BODY>
</HTML>
```

Wasn't THAT fun! No I suppose it wasn't. Take 5. Stretch, go pee, get cup of coffee, scratch yer butt, whatever. Take a break. I am.

Frames Tutor- Lesson 6

OK, let's make `home.html`, `here.html` and `there.html`. Since you've been working very hard (and I'm sick of typing), just go ahead and copy the following into Notepad and save them.

Save the following as `home.html`

```
<HTML>
<HEAD>
<TITLE>A Practice Page- Home</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN=center>You are home.</H1>
Welcome to my homepage! This page should be attractive and easy to navigate because my
teacher Joe, besides being VERY handsome, has taught me the right way to build Frames!<P>
<CENTER>Enjoy your stay!</CENTER><P>
```



```
</BODY>
</HTML>
```

[VIEW home.html](#)

Save this as `here.html`

```
<HTML>
<HEAD>
<TITLE>A Practice Page- Here</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN=center>Here</H1>
Welcome to here. Here is where everything happens. Here you'll find more neat stuff than you
can shake a stick at. Have you ever seen anyone shake a stick at something?? I never have.
</BODY>
</HTML>
```

[VIEW here.html](#)

And of course this one as `there.html`.

```
<HTML>
<HEAD>
<TITLE>A Practice Page- There</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN=center>This is There</H1>
I would like to make a proposal. We have all seen that little arrow on signs that say "You
are here". Wouldn't it be better if there were also another arrow that says "You should be
there". It would make things so much easier.
</BODY>
</HTML>
```

[VIEW there.html](#)

Now view your [Master Page](#).

We forgot something. If you caught the problem back when you were making the directory page... good! So open up `directry.html` and add the necessary TARGETS.

```
<HTML>
<HEAD>
<TITLE>Practice Page- Directory</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">

<H3>Directory</H3>
<A HREF="home.html" TARGET="MAIN-WINDOW">Home</A><P>
<A HREF="here.html" TARGET="MAIN-WINDOW">Go here</A><BR>
<A HREF="there.html" TARGET="MAIN-WINDOW">or there</A><P>
or visit<BR>
<A HREF="http://www.yahoo.com/" TARGET="_top">Yahoo</A><BR>
<A HREF="http://home.netscape.com/" TARGET="_top">Netscape</A>
</BODY>
</HTML>
```

Now try your [Master Page](#).

Works pretty good, doesn't it! Now just for giggles, let's add a link from your home page to me at Professional Web Design. Open up `home.html` and plug it in. Make sure you add the proper `TARGET` because although I really like you, I don't want to show up in *your* window.

```
<HTML>
<HEAD>
<TITLE>A Practice Page- Home</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN=center>You are home.</H1>
Welcome to my homepage! This page should be attractive and easy to navigate because my
teacher Joe, besides being VERY handsome, has taught me the right way to build Frames!<P>
<CENTER>Enjoy your stay!</CENTER><P>
Visit Joe at <A HREF="http://junior.apk.net/~jbarta/" TARGET="_top">Professional Web
Design</A>. </BODY>
</HTML>
```

View the [finished product](#).

Since first writing Frames Tutor, I have gotten many letters from people asking how to load *two* (or more) frames with one mouse click. This is reasonably easy to do with a little bit of JavaScript code. Since it's a "special occasion" kind of thing, and not normally used to develop framed pages, I'm keeping it kinda sorta separate. Unless you are specifically looking to add this feature to your pages, feel free to skip it.

[*How to load two \(or more\) frames with one click.*](#)

Frames Tutor- Lesson 7 (last one!)

Well folks, that's it!

For this tutorial that is. You have learned everything you need to create Framed pages. There is a little more, but 95% of what you'll need to do can be done with what was covered in this tutorial..

Form Tutor

Hello. My name is IT Kids and I'm going to show you how to add interactivity to your web documents by way of the `<FORM>` tag. With the form tag you can add to your web pages a guestbook, order forms, surveys, get feedback or whatever.

The basic construction of a html form is this...

<code><FORM></code>	begin a form
<code><INPUT></code>	ask for information in one of several different ways...
<code><INPUT></code>	...there can be as many input areas as you wish
<code></FORM></code>	end a form

That's html forms in a nutshell. You are now ready to make some forms! Now's a good time to stress that if you want to learn how to make quality html documents, then you would be well served to take the time to teach yourself the tags. If you rely on the so-called "form wizards" in the "easy as pie html editors" out there, you will have greatly limited flexibility, and the end result may not be what you are trying to achieve. In my opinion the best html editors to use are text based editors. . They don't attempt to do it for you.

Although this tutorial is not "optimised" for any particular browser, I wrote it using Netscape Navigator. If you are using MS Internet Explorer, some elements may not be supported or may be rendered a little differently.

Form Tutor- Lesson 1

Just like "follow the bouncing ball", I want you to open up Notepad (Yes Notepad!) and follow me. Copy and paste off this page the following to get you started:

```
<HTML>
<HEAD>
<TITLE>Joe's the handsomest guy I know</TITLE>
</HEAD>
<BODY>

</BODY>
</HTML>
```

Save it as `form1.html` in some folder somewhere (Win3.x users save it as `form1.htm`). Go ahead and give it its own folder. Start up your browser [[plug Netscape](#)]. Use it to open `form1.html` and run Notepad and the browser side by side. This way you can create your pages and almost instantaneously see the results of your handiwork. If hitting the reload button is not quite resetting everything, hit the reload button while holding down the [Shift] key.

Type in your form tags.

```
<HTML>
<HEAD>
<TITLE>Joe's the handsomest guy I know</TITLE>
</HEAD>
<BODY>
```

```
<FORM>
</FORM>
</BODY>
</HTML>
```

Next we must tell the browser where to send the data we gather and how to send it. There are two basic things you can do. 1) you can send the data to a cgi script for processing, or 2) you can have the data emailed to you. As for the first, whoever wrote the script can tell you how the data should be sent.

The second, or *mailto* form should have the following attributes in the `<FORM>` tag.

Note- Microsoft's Internet Explorer 3.0 does not support mailto forms. When you try to submit the information, the new mail message window pops up. Explorer does however support forms sent to a CGI script.

```
<HTML>
<HEAD>
<TITLE>Joe's the handsomest guy I know</TITLE>
</HEAD>
<BODY>

<FORM METHOD=POST ACTION="mailto:xxx@xxx.xxx" ENCTYPE="application/x-www-form-urlencoded">
</FORM>
</BODY>
</HTML>
```

This line is very important. The only thing you have to do is plug in your email address after mailto: The rest must be written exactly as shown. The words FORM, METHOD, POST & ACTION do not have to be capitalized *but* there must be a space between each attribute.. between FORM & METHOD, between POST & ACTION, and between `.com` & ENCTYPE.

Unfortunately the data will be sent to you in this 'only useful to a computer' format...

```
FORMNAME=New+Entrant&NAME=R.U.+Havinfun&ADDRESS=1313+Mockingbird+Lane
&CITY=Beverly+Hills&STATE=CA
```

What you'll need is a program to turn it into 'useful to a human' format...

```
FORMNAME=New Entrant  
NAME=R.U. Havinfun  
ADDRESS=1313 Mockingbird Lane  
CITY=Beverly Hills  
STATE=CA
```

[Mailto Formatter](#) is an excellent little freeware utility that does this job quite nicely.

The example above illustrates that a form is nothing more than input *names* (NAME, ADDRESS, etc) paired with input *values* (R.U. Havinfun, 1313 Mockingbird Lane, etc). The only real variable is *how* we go about getting the values.

Some mail programs are capable of converting the data without resorting to a separate program. You may want to try this method first. Just remove the instruction `ENCTYPE="application/x-www-form-urlencoded"` and in its place use `ENCTYPE="text/plain"`.

Form Tutor- Lesson 2

Now, just to keep things a little cleaner I am going to start writing only what is *within* the <FORM> tags. I will leave out the head, body, title and form tags from now on. Needless to say, leave them in your document.

The most common TYPE of form <INPUT> is TEXT.

```
<INPUT TYPE=TEXT>
```

Every input needs a NAME.

```
<INPUT TYPE=TEXT NAME="ADDRESS">
```

When the user types in his address (for example 1313 Mockingbird Lane), it will become the input's *value* and be paired with ADDRESS so the end result after running it through Mailto Formatter will be ADDRESS=1313 Mockingbird Lane.

We can if we want, type in a VALUE.

```
<INPUT TYPE=TEXT NAME="ADDRESS" VALUE="44 Cherry St">
```

This will automatically pair the value 44 Cherry St with the name ADDRESS, *unless* the user changes it. Note- be sure to use quotes where I've specified.

We can specify the size of the text input box.

```
<INPUT TYPE=TEXT NAME="ADDRESS" VALUE="44 Cherry St" SIZE=10>
```

```
<INPUT TYPE=TEXT NAME="ADDRESS" VALUE="44 Cherry St" SIZE=20>
```

```
<INPUT TYPE=TEXT NAME="ADDRESS" VALUE="44 Cherry St" SIZE=30>
```

As you can see, the default value is 20. You probably already know, by the way, that the *default value* is the value that the browser assumes if you have not told it otherwise.

Go ahead and remove `VALUE="44 Cherry St"`.

```
<INPUT TYPE=TEXT NAME="ADDRESS" SIZE=30>
```

If we want, we can specify how many characters a user can input. Just go ahead and *try* to input more than 10 characters!

```
<INPUT TYPE=TEXT NAME="ADDRESS" SIZE=30 MAXLENGTH=10>
```

I suppose this feature might come in handy now and again, but unless you think someone's going to send the whole King James Bible down the pike at you, I wouldn't worry about it.

Very similar to the `TYPE=TEXT` is the `TYPE=PASSWORD`. It is exactly the same, except it displays `***` instead of the actual input. The browser will *send* you the input, it just won't *display* it.

```
<INPUT TYPE=PASSWORD>
```

Remember that each `<INPUT>` must have a `NAME`.

```
<INPUT TYPE=PASSWORD NAME="USER PASSWORD">
```

`SIZE`, `VALUE`, and `MAXLENGTH` attributes work here also. By the way, a `<TAG>` tells the browser to do something. An `ATTRIBUTE` goes inside the `<TAG>` and tells the browser *how* to do it.

Next up are Radio Buttons and Check Boxes. Radio buttons allow the user to choose one of several options. Check Boxes allow the user to choose one or more or *all* of the options.

First let's build some Radio Buttons.

```
<INPUT TYPE=RADIO NAME="BEST FRIEND">
```



Now add 2 more.

```
<INPUT TYPE=RADIO NAME="BEST FRIEND">
<INPUT TYPE=RADIO NAME="BEST FRIEND">
<INPUT TYPE=RADIO NAME="BEST FRIEND">
```



Hmmm... I suppose we should put a line break after each one.

```
<INPUT TYPE=RADIO NAME="BEST FRIEND"><BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND"><BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND"><P>
```



Note that each input has the same name. The reason will become apparent very shortly.

Each of the Radio Buttons must be assigned a VALUE.

```
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Ed"><BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Rick"><BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Tom"><P>
```



Now label each button.

```
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Ed"> Ed Holleran<BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Rick"> Rick Weinberg<BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Tom"> Tom Studd<P>
```

- ☒ Ed Holleran
- ☐ Rick Weinberg
- ☐ Tom Studd

You can also modify these labels with other html tags if you wish.

Essentially your Radio Buttons are done. You can tidy things up by adding a statement above the buttons, and if you want, choose a default selection (optional).

```
Who is your best friend?<BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Ed" CHECKED> Ed Holleran<BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Rick"> Rick Weinberg<BR>
<INPUT TYPE=RADIO NAME="BEST FRIEND" VALUE="Tom"> Tom Studd<P>
```

Who is your best friend?

- ☒ Ed Holleran
- ☐ Rick Weinberg
- ☐ Tom Studd

The user of course can only choose 1 option. Their choice will be returned to you as the name/value pair BEST FRIEND=Ed (or whoever they pick).

Building Check Boxes is pretty much the same thing. Start with this.

```
<INPUT TYPE=CHECKBOX NAME="Ed">
```

☐

Add 3 more, but this time give each one a different NAME. (Also add in line breaks if you want)

```
<INPUT TYPE=CHECKBOX NAME="ED"><BR>
<INPUT TYPE=CHECKBOX NAME="Rick"><BR>
<INPUT TYPE=CHECKBOX NAME="Tom"><BR>
<INPUT TYPE=CHECKBOX NAME="BM"><P>
```

☐☐☐☐

Each Check Box gets the same VALUE.

```
<INPUT TYPE=CHECKBOX NAME="ED" VALUE="YES"><BR>
<INPUT TYPE=CHECKBOX NAME="Rick" VALUE="YES"><BR>
```

```
<INPUT TYPE=CHECKBOX NAME="Tom" VALUE="YES"><BR>
<INPUT TYPE=CHECKBOX NAME="BM" VALUE="YES"><P>
```

☐
☐
☐
☐

Note- For Check Boxes the NAME changes and the VALUE stays the same and with Radio Buttons, the VALUE changes but the NAME stays the same. Don't feel bad, my simple mind still gets confused. That's why I lean **heavily** on html reference documents. (You thought I had all this in my head?? HA!)

OK, let's label each box.

```
<INPUT TYPE=CHECKBOX NAME="ED" VALUE="YES"> Ed Holleran<BR>
<INPUT TYPE=CHECKBOX NAME="Rick" VALUE="YES"> Rick Weinberg<BR>
<INPUT TYPE=CHECKBOX NAME="Tom" VALUE="YES"> Tom Studd<BR>
<INPUT TYPE=CHECKBOX NAME="BM" VALUE="YES"> Burgermeister Meisterburger<P>
```

☐ Ed Holleran
☐ Rick Weinberg
☐ Tom Studd
☐ Burgermeister Meisterburger

And lastly, you may want to add a little something above your check boxes and maybe pick a couple defaults. Only if you want to, of course.

```
Which of these guys are your friends?<BR>
<INPUT TYPE=CHECKBOX NAME="ED" VALUE="YES" CHECKED> Ed Holleran<BR>
<INPUT TYPE=CHECKBOX NAME="Rick" VALUE="YES"> Rick Weinberg<BR>
<INPUT TYPE=CHECKBOX NAME="Tom" VALUE="YES" CHECKED> Tom Studd<BR>
<INPUT TYPE=CHECKBOX NAME="BM" VALUE="YES"> Burgermeister Meisterburger<P>
```

Which of these guys are your friends?

☒ Ed Holleran
☐ Rick Weinberg
☒ Tom Studd
☐ Burgermeister Meisterburger

The user can choose 1, 2, none or all of the options. Their choices will be returned to you as the name/value pairs...

Ed=YES
Tom=YES

(or what ever they choose... if they choose nothing, nothing will be returned to you)

Now a question might come to mind... What if I want to ask 3 different questions about the same group of guys?? How, Mr Smartypants am I going to do that!

Well, just settle down and I'll show you.

Which of these guys are your friends? <input type="checkbox"/> Ed Holleran <input type="checkbox"/> Rick Weinberg <input type="checkbox"/> Tom Studd <input type="checkbox"/> Burgermeister	Which of these guys would you lend money to? <input type="checkbox"/> Ed Holleran <input type="checkbox"/> Rick Weinberg <input type="checkbox"/> Tom Studd <input type="checkbox"/> Burgermeister	Which of these guys would you trust with your sister? <input type="checkbox"/> Ed Holleran <input type="checkbox"/> Rick Weinberg <input type="checkbox"/> Tom Studd <input type="checkbox"/> Burgermeister
---	--	---

It's true that in each form there should never be duplicate NAMES. So, maybe we could use a *different name* for each question. When I say never, I don't mean that your computer will blow up... at most it might confuse the browser, or the parser (Mailto Formatter is a parser), or the cgi script. At the least it will confuse the poor sap that has to make sense of the form data.

What follows is the html for these 3 questions. The <TABLE> tags are in green. They are for appearance only, they don't affect how the form works. If you need to brush up on your <TABLE> tags, then stumble on over to [Table Tutor](#).

```
<CENTER>
<TABLE WIDTH=600 BORDER=1 CELLSPACING=1><TR>

<TD WIDTH=199>
Which of these guys are your friends?<BR>
<INPUT TYPE=CHECKBOX NAME="Friend?..Ed" VALUE="YES"> Ed Holleran<BR>
<INPUT TYPE=CHECKBOX NAME="Friend?..Rick" VALUE="YES"> Rick Weinberg<BR>
<INPUT TYPE=CHECKBOX NAME="Friend?..Tom" VALUE="YES"> Tom Studd<BR>
<INPUT TYPE=CHECKBOX NAME="Friend?..BM" VALUE="YES"> Burgermeister<P>
</TD>
<TD WIDTH=200>
Which of these guys would you lend money to?<BR>
<INPUT TYPE=CHECKBOX NAME="Lend money?...Ed" VALUE="YES"> Ed Holleran<BR>
<INPUT TYPE=CHECKBOX NAME="Lend money?...Rick" VALUE="YES"> Rick Weinberg<BR>
<INPUT TYPE=CHECKBOX NAME="Lend money?...Tom" VALUE="YES"> Tom Studd<BR>
<INPUT TYPE=CHECKBOX NAME="Lend money?...BM" VALUE="YES"> Burgermeister<P>
</TD>
<TD WIDTH=199>
Which of these guys would you trust with your sister?<BR>
<INPUT TYPE=CHECKBOX NAME="Date sister?...Ed" VALUE="YES"> Ed Holleran<BR>
<INPUT TYPE=CHECKBOX NAME="Date sister?...Rick" VALUE="YES"> Rick Weinberg<BR>
<INPUT TYPE=CHECKBOX NAME="Date sister?...Tom" VALUE="YES"> Tom Studd<BR>
<INPUT TYPE=CHECKBOX NAME="Date sister?...BM" VALUE="YES"> Burgermeister<P>
</TD>
</TR></TABLE>
</CENTER>
```

Let's suppose the user checked the following boxes...

Which of these guys are your friends? <input checked="" type="checkbox"/> Ed Holleran <input checked="" type="checkbox"/> Rick Weinberg <input checked="" type="checkbox"/> Tom Studd <input type="checkbox"/> Burgermeister	Which of these guys would you lend money to? <input type="checkbox"/> Ed Holleran <input type="checkbox"/> Rick Weinberg <input checked="" type="checkbox"/> Tom Studd <input checked="" type="checkbox"/> Burgermeister	Which of these guys would you trust with your sister? <input checked="" type="checkbox"/> Ed Holleran <input checked="" type="checkbox"/> Rick Weinberg <input type="checkbox"/> Tom Studd <input checked="" type="checkbox"/> Burgermeister
--	--	--

...doing that would send you the following name/value pairs.

```
Friend?..Ed=YES
Friend?..Rick=YES
Friend?..Tom=YES
Lend money?...Tom=YES
Lend money?...BM=YES
Date sister?...Ed=YES
Date sister?...Tom=YES
Date sister?...BM=YES
```

Isn't that cool!

Form Tutor- Lesson 3

Before we go on, there is one more thing I want to mention. You can also use *images* in a form. As a matter of fact, you can use just about anything in a form or a form in anything. Just watch your html syntax and avoid overlapping tags.

Overlapping tags, for those that are wondering are tags that... um... overlap. Let me illustrate.

```
<TABLE><FORM></TABLE></FORM> Overlapping tags.... bad
<TABLE><FORM></FORM></TABLE> Nested tags.... good
```

The next type of input is a Pull Down List. With this type you use `<SELECT>` instead of `<INPUT>` and it has a closing tag. Let's make one.

```
<SELECT>
</SELECT>
```



Don't forget to give it a name.

```
<SELECT NAME="BEST_FRIEND">
</SELECT>
```



Next add a few *options*.

```
<SELECT NAME="BEST_FRIEND">
<OPTION>Ed
<OPTION>Rick
<OPTION>Tom
<OPTION>Guido
</SELECT>
```



And give each <OPTION> a VALUE.

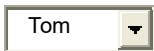
```
<SELECT NAME="BEST FRIEND">
<OPTION VALUE="Ed">Ed
<OPTION VALUE="Rick">Rick
<OPTION VALUE="Tom">Tom
<OPTION VALUE="Guido">Guido
</SELECT>
```

A pull-down menu with a light gray border. The text 'Ed' is displayed in the main area, and a small downward-pointing arrow is visible on the right side of the menu.

The default option is the one that is listed first.

We can specify a default other than the first option in the list.

```
<SELECT NAME="BEST FRIEND">
<OPTION VALUE="Ed">Ed
<OPTION VALUE="Rick">Rick
<OPTION VALUE="Tom" SELECTED>Tom
<OPTION VALUE="Guido">Guido
</SELECT>
```

A pull-down menu with a light gray border. The text 'Tom' is displayed in the main area, and a small downward-pointing arrow is visible on the right side of the menu.

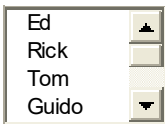
A Scrolling List is very similar in construction to a Pull Down List . Let's add a few more names first though.

```
<SELECT NAME="BEST FRIEND">
<OPTION VALUE="Ed">Ed
<OPTION VALUE="Rick">Rick
<OPTION VALUE="Tom">Tom
<OPTION VALUE="Guido">Guido
<OPTION VALUE="Horace">Horace
<OPTION VALUE="Reggie">Reggie
<OPTION VALUE="Myron">Myron
</SELECT>
```

A pull-down menu with a light gray border. The text 'Ed' is displayed in the main area, and a small downward-pointing arrow is visible on the right side of the menu.

All we gotta do to turn it into a Scrolling List is add a SIZE attribute to the <SELECT> tag.

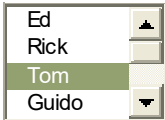
```
<SELECT NAME="BEST FRIEND" SIZE=4>
<OPTION VALUE="Ed">Ed
<OPTION VALUE="Rick">Rick
<OPTION VALUE="Tom">Tom
<OPTION VALUE="Guido">Guido
<OPTION VALUE="Horace">Horace
<OPTION VALUE="Reggie">Reggie
<OPTION VALUE="Myron">Myron
</SELECT>
```



The SIZE is simply how many options show in the window. This stuff is CAKE!

Again, the default value is the first <OPTION>, and again we can change that by selecting one.

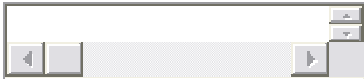
```
<SELECT NAME="BEST FRIEND" SIZE=4>
<OPTION VALUE="Ed">Ed
<OPTION VALUE="Rick">Rick
<OPTION VALUE="Tom" SELECTED>Tom
<OPTION VALUE="Guido">Guido
<OPTION VALUE="Horace">Horace
<OPTION VALUE="Reggie">Reggie
<OPTION VALUE="Myron">Myron
</SELECT>
```



I have no idea *why* you would use the selection feature for a Scrolling List but it's there and I just felt the need to tell you about it. 😊

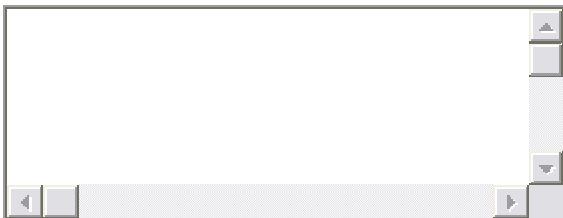
A very useful type of input is <TEXTAREA>.

```
<TEXTAREA NAME="COMMENTS">
</TEXTAREA>
```

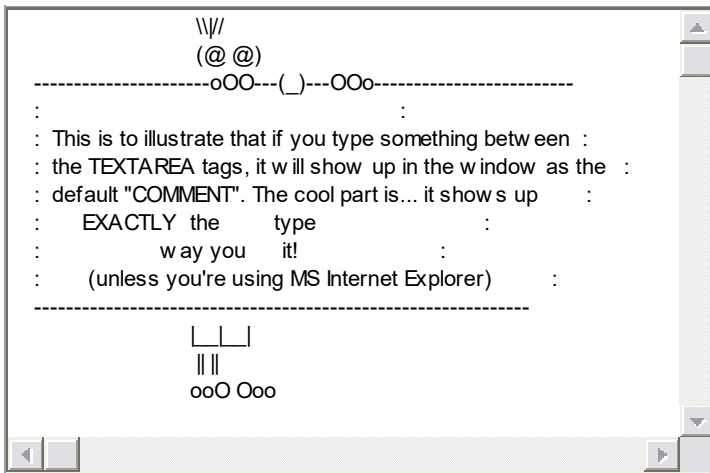


You control the size of the box like so...

```
<TEXTAREA NAME="COMMENTS" ROWS=6 COLS=50>
</TEXTAREA>
```



ROWS is the height, COLS is the width.



Html code for this textarea [here](#).

A good attribute to include in <TEXTAREA> is WRAP. Some browsers do not understand it, but if that's the case, they will just ignore it.

Go ahead and type in the boxes...

```
<TEXTAREA NAME="COMMENTS" ROWS=3 COLS=30 WRAP=VIRTUAL>
</TEXTAREA>
```

WRAP=VIRTUAL means that the text in the box wraps, but it is *sent* as one long continuous string.

```
<TEXTAREA NAME="COMMENTS" ROWS=3 COLS=30 WRAP=PHYSICAL>
</TEXTAREA>
```

WRAP=PHYSICAL means that the text in the box wraps, *and* it is *sent* that way too.

```
<TEXTAREA NAME="COMMENTS" ROWS=3 COLS=30 WRAP=OFF>
</TEXTAREA>
```

This is the default.

WRAP=OFF means that the text in the box *does not* wrap, but it is *sent* exactly the way it was typed in (like the little man a few textareas back).

Form Tutor- Lesson 4

Yet another type of input is HIDDEN input.

```
<INPUT TYPE=HIDDEN NAME="FORMNAME" VALUE="Friend Form 1">
```

A HIDDEN input is a name/value pair that is returned to you but does not show up anywhere on the web page. The hidden input above is needed for use with Mailto Formatter. It is how MTF recognizes the forms it's supposed to parse.

Let's suppose you were a company trying to generate leads for a new product. You have a standard form for gathering information... name, company, phone, products interested in, etc. The only problem is there are 6 slightly different versions of the form in 6 slightly different places. You need to know what's coming from where. What to do?

You could add a HIDDEN input to your forms like so...

```
<INPUT TYPE=HIDDEN NAME="FORMNAME" VALUE="Version 1"> ...for the first version
<INPUT TYPE=HIDDEN NAME="FORMNAME" VALUE="Version 2"> ...for the second version
<INPUT TYPE=HIDDEN NAME="FORMNAME" VALUE="Version 3"> ...for the third version
And so on and so forth yada yada yada.
```

By the way, it doesn't matter what the name/value pair in the hidden input is (or *any* input for that matter). I have just been using "FORMNAME" because it saved me some typing. This would be a perfectly legitimate HIDDEN input...

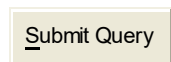
```
<INPUT TYPE=HIDDEN NAME="E" VALUE="Mc^2"> ...You would get back E=Mc^2
```

HIDDEN inputs are also useful for cgi scripts. For example, many Internet Service Providers have a script you can have your forms sent to. It then spits the form back to you all nice and neat and ready for human consumption. The hidden input tells the cgi script who you are, where to send the parsed data, etc.

Last on the list are the SUBMIT and RESET buttons.

They really are very simple...

```
<INPUT TYPE=SUBMIT>
```



SUBMIT of course, sends the data...

...and RESET, clears the form.

```
<INPUT TYPE=RESET>
```



We can easily change what the buttons say.

```
<INPUT TYPE=SUBMIT VALUE="Send it away Ray!"><BR>
<INPUT TYPE=RESET VALUE="Clear the form Norm!"><P>
```

The image shows two buttons stacked vertically. The top button is light yellow with a thin black border and contains the text "Send it away Ray!". The bottom button is also light yellow with a thin black border and contains the text "Clear the form Norm!".

If necessary, the `SUBMIT` button can also have a `NAME`. You would need this if, for whatever reason, you had more than one `SUBMIT` button.

One more little tidbit and we're going to wrap this up. When you put a `mailto` form on your page and someone sends you information, you'll notice that it is sent with a default Subject. If you're visitor was using Netscape you'd get the default Subject "*Form posted from Mozilla*". Other browsers might send "*Form Response*", etc.

You can change this by editing what's in the `<FORM>` tag as follows...

```
<FORM METHOD=POST ACTION="mailto:michael@corleone.com?subject=Our friends in Las Vegas"
ENCTYPE="application/x-www-form-urlencoded">
```

Pretty cool huh?

TABLE TUTOR

The whole mess basically has only 3 tags...

<TABLE>

The main tag. Used to tell the browser "this is a table", along with some attributes like size, border width and a few other things.

<TR>

TableRow defines a horizontal row of <TD> (TableData) *cells*.

<TD>

Specifies an individual block or *cell* in a table row.

To paraphrase: A *table* is made up of *rows* which in turn are made up of *cells*.

Table Tutor - Lesson 1

Just like "follow the bouncing ball", I want you to open up Notepad (Yes Notepad!) and follow me. Copy and paste off this page the following to get you started:

```
<HTML>
<HEAD>
<TITLE>Joe's a swell guy</TITLE>
</HEAD>
<BODY>

</BODY>
</HTML>
```

Save it as `table1.html` in some folder somewhere (Win3.x users save it as `table1.htm`). Go ahead and give it its own folder because we will be putting other things in there too. Start up your browser [[plug Netscape](#)]. Use it to open `table1.html` and run Notepad and the browser side by side. This way you can create your pages and almost instantaneously see the results of your handiwork. And you thought you needed a 9 megabyte 'wizzard'.

Type in your table tags. These simply mean "starting a table" and "ending a table".

```
<HTML>
<HEAD>
<TITLE>Joe's a swell guy</TITLE>
</HEAD>
<BODY>

<TABLE>
</TABLE>
</BODY>
</HTML>
```

Every table needs at least one row.

```
<HTML>
<HEAD>
<TITLE>Joe's a swell guy</TITLE>
</HEAD>
<BODY>

<TABLE>
<TR>
</TR>
</TABLE>
</BODY>
</HTML>
```

And of course every row has to have at least one table data *cell*.

```
<HTML>
<HEAD>
<TITLE>Joe's a swell guy</TITLE>
</HEAD>
<BODY>

<TABLE>
<TR>
<TD></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Now, just to keep things a little cleaner I am going to start writing only what is in the table tags. I will leave out the head, body, title, etc. tags from now on. Needless to say, leave them in your document.

```
<TABLE>
<TR>
<TD></TD>
</TR>
</TABLE>
```


Now you need something to put in that cell. How about Ed? let's put Ed in the <TD> cell.

```
<TABLE>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Well now, you are the proud owner of a fully functional html table! Open it with the browser and check it out! If you've done everything correctly your table should look like this:

Ed		

Just kidding! Haha! As you can see I am easily amused. This is what you have created...

Ed

Take a deep breath, pat yourself on the back and be proud. You made your first table!

Table Tutor - Lesson 2

Ok, what we've got is this...

```
<TABLE>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

...which gives us this:

Ed

First lets make it look more like a table and give it a border. Every time you make a change and want to see how it looks, you can hit the Reload button on your browser. If your browser doesn't have a reload button, [get another browser](#).

```
<TABLE BORDER=1>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

How about a bigger border?

```
<TABLE BORDER=5>
<TR>
<TD>Ed</TD>
```

```
</TR>
</TABLE>
```

Ed

How about a HUGE border?

```
<TABLE BORDER=25>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

How about NO border?

```
<TABLE BORDER=0>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

As you can see, the default is no border.

Let's stick with a modest border for now.

```
<TABLE BORDER=3>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

When no sizes are specified the table is only as big as it needs to be.

```
<TABLE BORDER=3>
<TR>
<TD>Ed, Rick and Tom</TD>
</TR>
</TABLE>
```

Ed, Rick and Tom

Specifying a table size is pretty simple though.

```
<TABLE BORDER=3 WIDTH=100%>
<TR>
<TD>Ed, Rick and Tom</TD>
</TR>
</TABLE>
```

Ed, Rick and Tom

How about this?

```
<TABLE BORDER=3 WIDTH=75%>
<TR>
<TD>Ed, Rick and Tom</TD>
</TR>
</TABLE>
```

Ed, Rick and Tom

Let's get rid of Ed's friends for now.

```
<TABLE BORDER=3 WIDTH=75%>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

Let's reduce the table size to 50% of the browser window.

```
<TABLE BORDER=3 WIDTH=50%>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

Now we will specify a width of 50 instead of 50%.

```
<TABLE BORDER=3 WIDTH=50>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

Now 100.

```
<TABLE BORDER=3 WIDTH=100>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

As you can see there are two ways to specify table width. Each style has its uses. Which style to use for what is unimportant now.

We can also mess around with the *height*.

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

I'm reminded of something Steven Wright said...

"The other day, I was walking my dog around my building...on the ledge. Some people are afraid of heights. Not me, I'm afraid of widths."

We can control where in the cell the data will appear.

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD ALIGN=CENTER>Ed</TD>
</TR>
</TABLE>
```

Ed

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
```

```
<TD ALIGN=RIGHT>Ed</TD>
</TR>
</TABLE>
```

Ed

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD ALIGN=LEFT>Ed</TD>
</TR>
</TABLE>
```

Ed

As you can see, the default value is `ALIGN=LEFT`. You probably already know, by the way, that the *default value* is the value that the browser assumes if you have not told it otherwise.

We can also control where data will appear *vertically* in a cell.

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD ALIGN=LEFT VALIGN=TOP>Ed</TD>
</TR>
</TABLE>
```

Ed

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD ALIGN=LEFT VALIGN=BOTTOM>Ed</TD>
</TR>
</TABLE>
```

Ed

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD ALIGN=LEFT VALIGN=MIDDLE>Ed</TD>
</TR>
</TABLE>
```

Ed

Default is middle.

I will take a second now to show you something else. Images can also be used and manipulated in a table data cell. In the folder that contains this document you will find a small picture named `ed.gif`. Copy it into the folder that you are working out of (the one where you put `tables1.html`). Substitute an `IMG` tag for `Ed`.

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD ALIGN=LEFT VALIGN=MIDDLE><IMG
SRC="ed.gif" WIDTH=32 HEIGHT=32></TD>
```

```
</TR>
</TABLE>
```



I suppose that now is also a good time to stress that it is good practice to include size attributes with all your image tags. I won't go into details, but doing so makes it easier for the browser and avoids any nasty little surprises.

Table Tutor - Lesson 3

Let's go back to plain old Ed.

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD ALIGN=LEFT VALIGN=MIDDLE>Ed</TD>
</TR>
</TABLE>
```



And for the sake of clarity and simplicity let's remove the alignment attributes. We know what will happen because we know what the default values are. By the way, a TAG tells the browser to do something. An ATTRIBUTE goes inside the TAG and tells the browser *how* to do it.

```
<TABLE BORDER=3 WIDTH=100 HEIGHT=75>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```



Now we will make our table a fuzz bigger.

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```



You should know that *fuzz* is a technical term. Its full definition is so broad and complicated that it would only be suitable for advanced html students.

Ed's friend Tom is back and this time he wants his own cell.

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>
<TR>
<TD>Ed</TD>
<TD>Tom</TD>
</TR>
</TABLE>
```



When no instructions are given to the browser, each cell may (but not always) be different in size. It's always a fine idea to specify how big each cell is. Make sure your arithmetic is correct or what people see may be drastically different than what you want them to see!

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>
<TR>
<TD WIDTH=150>Ed</TD>
<TD WIDTH=150>Tom</TD>
</TR>
</TABLE>
```



These WIDTH attributes can also be expressed as a percentage.

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>
<TR>
<TD WIDTH=50%>Ed</TD>
<TD WIDTH=50%>Tom</TD>
</TR>
</TABLE>
```



Lets give Ed a bigger cell since he's been here from the beginning.

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>
<TR>
<TD WIDTH=80%>Ed</TD>
<TD WIDTH=20%>Tom</TD>
</TR>
</TABLE>
```



Now Rick is back and of course he wants his own cell. We need to decide how much of the row we will give him. I suppose 20% is fair. Make sure to adjust Ed's share also.

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>
<TR>
<TD WIDTH=60%>Ed</TD>
<TD WIDTH=20%>Tom</TD>
<TD WIDTH=20%>Rick</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
----	-----	------

Three yahoos from across the street see what's going on and they want to be in your table. I think we will give them their own row.

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>

<TR>
<TD WIDTH=60%>Ed</TD>
<TD WIDTH=20%>Tom</TD>
<TD WIDTH=20%>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

The WIDTH attributes in the first row carry over to the second row.

If Mo leaves, we still have a perfectly good table, it just has an empty spot.

```
<TABLE BORDER=3 WIDTH=300 HEIGHT=75>

<TR>
<TD WIDTH=60%>Ed</TD>
<TD WIDTH=20%>Tom</TD>
<TD WIDTH=20%>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD></TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	

Let's put Mo back and remove all attributes except BORDER.

```
<TABLE BORDER=3>
```

```
<TR>
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

Next are a couple of attributes called CELLPADDING and CELLSPACING. Both are used up front in the <TABLE> tag. CELLPADDING is the amount of space between the border of the cell and the contents of the cell.

```
<TABLE BORDER=3 CELLPADDING=12>

<TR>
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

The default value for this attribute is 1. The reason it is 1 and not 0 is so that any text in the cells won't be banging up against the borders. (Although you could specify 0 if you wanted to).

If we substitute CELLSPACING for CELLPADDING we get a slightly different effect.

```
<TABLE BORDER=3 CELLSPACING=12>

<TR>
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
```

```
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

The default value for the CELSPACING attribute is 2.

We can, of course, use these attributes in combination.

```
<TABLE BORDER=3 CELSPACING=12
CELLPADDING=12>
```

```
<TR>
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

Table Tutor - Lesson 4

Let's get rid of CELLPADDING and CELSPACING and go back to our simple little table.

```
<TABLE BORDER=3>
```

```
<TR>
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

A cool feature of the newer browsers is the ability to specify background colors for a table cell, row or the whole table. You use BGCOLOR just like you would in the <BODY> tag.

```
<TABLE BORDER=3 BGCOLOR="#FFCC66">
```

```
<TR>
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

```
<TABLE BORDER=3>
<TR BGCOLOR="#FF9999">
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR BGCOLOR="#99CCCC">
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

```
<TABLE BORDER=3>
<TR BGCOLOR="#FFCCFF">
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD BGCOLOR="#FF0000">Larry</TD>
<TD>Curly</TD>
<TD BGCOLOR="#3366FF">Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

[Here's a handy chart](#) I use to choose background colors. You can also use [Color Picker](#).

One more thing about these table background colors... a <TD> bgcolor will override a <TR> bgcolor and a <TR> bgcolor will override a <TABLE> bgcolor. Not that this needs further explanation but I'm kind of having fun with this so here's an example:

```
<TABLE BORDER=3 BGCOLOR="#FF6633">
```

```
<TR BGCOLOR="#009900">
<TD BGCOLOR="#9999FF">Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

Let me take a minute to explain something. A browser has to interpret the instructions you give it the best way it can. If something has not been specified one way or another, most browsers will try to come up with an attractive solution. The best thing you can do as an author is to specify as much as you can, especially those things that are important for your page to look right. It is also important to view your work through those browsers that people actually use. Since most people use Netscape, that is a good start. You may also want to have a copy of a couple other popular browsers to be sure that you look right to them too.

Another consideration is screen resolution. I work on a 640x480 screen. Many people use 800x600 and a few have theirs set to 1024x768. This simple difference has the potential to seriously mess with your page design. There is a little utility that I use called [QuickRes](#) that can change your screen resolution back and forth with the click of a button. It's not a bad idea to view your pages through other resolutions.

Now we will play with COLSPAN (Column Span) and ROWSPAN (Row Span maybe??). Let's suppose Ed beats the crap out of Tom and throws him out of the table. Just doing that, this is what we have.

```
<TABLE BORDER=3>
```

```
<TR>
<TD>Ed</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Rick	
Larry	Curly	Mo

It just left an empty spot and Rick slid over to fill the void.

If we want Ed to actually take possession of Tom's cell and make the area part of his own, we have to use the COLSPAN attribute thusly.

```
<TABLE BORDER=3>
```

```
<TR>
<TD COLSPAN=2>Ed</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Rick	
Larry	Curly	Mo

To emphasise the point I made earlier, about the browser trying to find an attractive solution, let's make Ed span two columns *but we'll put Tom back in*. We will deliberately introduce a discrepancy just to see how the browser handles it.

```
<TABLE BORDER=3>
```

```
<TR>
<TD COLSPAN=2>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>
```

Ed	Tom	Rick
Larry	Curly	Mo

The point is 1) The browser is very forgiving in that it does the best it can with what you give it. 2) It is very important to specify what is important and make sure there are no discrepancies or you may end up with a surprise. And 3) nothing you do with html will crash the other person's browser no matter how badly you may have messed it up, so don't worry about that. (Unfortunately the same cannot be said for JAVA and animated gifs.)

OK... let's take Tom back out (the poor sap).

```
<TABLE BORDER=3>
```

```

<TR>
<TD COLSPAN=2>Ed</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>

```

Ed	Rick	
Larry	Curly	Mo

Rick gets scared and he leaves. Ed takes over Rick's cell too, and just cause he's that way, he stands right in the middle of the cell.

```
<TABLE BORDER=3>
```

```

<TR>
<TD COLSPAN=3 ALIGN=CENTER>Ed</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>

```

Ed		
Larry	Curly	Mo

All other html coding, by the way, can be used in a cell. We'll make Ed as an example.

```
<TABLE BORDER=3>
```

```

<TR>
<TD COLSPAN=3
ALIGN=CENTER><STRONG>Ed</STRONG></TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>

```

Ed		
Larry	Curly	Mo

We'll make him a link to my homepage.

```
<TABLE BORDER=3>
```

```

<TR>
<TD COLSPAN=3 ALIGN=CENTER>
<A

```

```

HREF="http://junior.apk.net/~jbarta/">Ed</A
></TD>
</TR>

```

```

<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>

```

Ed		
Larry	Curly	Mo

Now get rid of all that stuff, bring back Tom and Rick and we'll dive into <ROWSPAN>.

```
<TABLE BORDER=3>
```

```

<TR>
<TD>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Larry</TD>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>

```

Ed	Tom	Rick
Larry	Curly	Mo

As you may have guessed, <ROWSPAN> is just like <COLSPAN> 'cept yall span rows instead of columns. (Not exactly brain surgery... is it?) If we remove Larry and let Ed take over his cell, this is the result.

```
<TABLE BORDER=3>
```

```

<TR>
<TD ROWSPAN=2>Ed</TD>
<TD>Tom</TD>
<TD>Rick</TD>
</TR>
<TR>
<TD>Curly</TD>
<TD>Mo</TD>
</TR>
</TABLE>

```

Ed	Tom	Rick
	Curly	Mo

And of course, these tags can also be used in combination.

```
<TABLE BORDER=3>
```



```

<TR>
<TD ROWSPAN=2>Ed</TD>
<TD COLSPAN=2>Tom</TD>
</TR>
<TR>
<TD>Curly</TD>

```

```

<TD>Mo</TD>
</TR>
</TABLE>

```

Ed	Tom
Curly	Mo

Table Tutor - Lesson 5

Now we're going to play around a little.. try a few things just to see what will happen.

Here's a little unordered list:

Ingredients for Apple Pie

- Apples
- Flour
- Sugar
- Cinnamon

That's all fine & dandy, but what if we want to put it over here...

Ingredients for Apple Pie

- Apples
- Flour
- Sugar
- Cinnamon

Because we want it to be next to a picture of... a pie!



Ingredients for Apple Pie

- Apples
- Flour
- Sugar
- Cinnamon

When studying or building a table, it's always easier when the borders are turned on.

Start with little Ed.

```

<TABLE BORDER=3>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>

```

Ed

Replace Ed with the unordered list.

```

<TABLE BORDER=3>
<TR>

<TD>
Ingredients for Apple Pie
<UL>
<LI>Apples
<LI>Flour
<LI>Sugar
<LI>Cinnamon
</UL>
</TD>
</TR>
</TABLE>

```

Ingredients for Apple Pie

- Apples
- Flour
- Sugar
- Cinnamon

Remember, in the absence of other instructions, the table will make itself just big enough to contain the data. So in this instance, a size attribute is not really needed.

Next we'll expand the table the full width of the browser window.

```

<TABLE BORDER=3 WIDTH=100%>
<TR>

<TD>
Ingredients for Apple Pie
<UL>
<LI>Apples
<LI>Flour
<LI>Sugar
<LI>Cinnamon
</UL>
</TD>
</TR>
</TABLE>

```

Ingredients for Apple Pie

- Apples
- Flour
- Sugar
- Cinnamon

Now we need to make a second cell. We want the left cell to be a little smaller than the right one.

```

<TABLE BORDER=3 WIDTH=100%>
<TR>

<TD WIDTH=40%></TD>
<TD WIDTH=60%>
Ingredients for Apple Pie
<UL>
<LI>Apples

```

```

<LI>Flour
<LI>Sugar
<LI>Cinnamon
</UL>
</TD>
</TR>
</TABLE>

```

	<h3>Ingredients for Apple Pie</h3> <ul style="list-style-type: none"> • Apples • Flour • Sugar • Cinnamon
--	---

Now would be a good time to copy `applepie.gif` to your working folder.
 Plug the image into the first cell.

```


<TABLE BORDER=3 WIDTH=100%>
<TR>

```

```

<TD WIDTH=40%><IMG SRC="applepie.gif" WIDTH=150 HEIGHT=138></TD>
<TD WIDTH=60%>
Ingredients for Apple Pie
<UL>
<LI>Apples
<LI>Flour
<LI>Sugar
<LI>Cinnamon
</UL>
</TD>
</TR>
</TABLE>

```

	<h3>Ingredients for Apple Pie</h3> <ul style="list-style-type: none"> • Apples • Flour • Sugar • Cinnamon
--	---

All that's left is to align the pie image to the right side of the cell and turn off the border attribute.

```

<TABLE BORDER=0 WIDTH=100%>
<TR>

```

```

<TD WIDTH=40% ALIGN=RIGHT><IMG SRC="applepie.gif" WIDTH=150 HEIGHT=138></TD>
<TD WIDTH=60%>
Ingredients for Apple Pie
<UL>
<LI>Apples
<LI>Flour
<LI>Sugar
<LI>Cinnamon
</UL>
</TD>
</TR>
</TABLE>

```



Ingredients for Apple Pie

- Apples
- Flour
- Sugar
- Cinnamon

Bingo!

Table Tutor - Lesson 6

Can we have a table *within* a table??
Let's try. Start with little Ed.

```
<TABLE BORDER=3>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

Make the table a *fuzz* bigger.

```
<TABLE BORDER=3 WIDTH=200 HEIGHT=100>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
```

Ed

Replace Ed with a complete "little Ed" table

```
<TABLE BORDER=3 WIDTH=200 HEIGHT=100>
<TR>
<TD>

<TABLE BORDER=3>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
```

Ed

Our table can also be centered very easily.

```
<CENTER>
<TABLE BORDER=3 WIDTH=200 HEIGHT=100>
<TR>
<TD>
```

```
<TABLE BORDER=3>
<TR>
<TD>Ed</TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
</CENTER>
```

Ed

Here's a situation where a table can solve a common problem.

Say you've got this great idea for an animated gif. But the picture you want to use is pretty big. Since you know that an animated gif is basically a series of gifs displayed one after another, you cringe at the size this thing is going to be. Your viewers won't be too happy about a 700K clodhopper coming down the pike at them. Not only that, but because of its size, their browser is going to choke trying to display it. The answer? Cut it up and display it as a *table*.

Here's an example:

Netscape vs. Microsoft?



Artwork by [Boris Vallejo](#)

The only part of the picture that is animated are the eyes.

Here is the html code for that table. (The caption is just something I added for my own amusement. It is not required nor does it have any impact on the rest of the table)

```
<TABLE WIDTH=591 BORDER=0 CELLSPACING=0 CELLPADDING=0>
<CAPTION ALIGN=top><FONT SIZE=6><STRONG><EM>Netscape vs.
Microsoft?</EM></STRONG></FONT></CAPTION>

<TR>
<TD ROWSPAN=3><IMG SRC="dragon/dragon4.gif" WIDTH=250 HEIGHT=406></TD>
<TD><IMG SRC="dragon/dragon3.gif" WIDTH=122 HEIGHT=109></TD>
<TD ROWSPAN=3><IMG SRC="dragon/dragon5.gif" WIDTH=219 HEIGHT=406></TD>
</TR>
<TR>
<TD><IMG SRC="dragon/anidrag.gif" WIDTH=122 HEIGHT=50></TD>
</TR>
```

```
<TR>
<TD><IMG SRC="dragon/dragon2.gif" WIDTH=122 HEIGHT=247></TD>
</TR>
</TABLE>
```

This is the layout.

dragon4.gif	dragon3.gif	dragon5.gif
	anidrag.gif	
	dragon2.gif	

Well that's about it for the table tags. Used to be that this was the end of the tutorial, but since I cannot stress enough the important role table tags play in the creation of high quality web pages, I've added a [few more examples](#). Try to reproduce them yourself.



***** INDEX OF TABLE TAGS AND ATTRIBUTES *****
(the ones that are explained in this tutorial)

[ALIGN](#) *As used in <TD>*

[BGCOLOR](#)

[BORDER](#)

[CELLPADDING](#)

[CELLSPACING](#)

[<CENTER>](#)

[COLSPAN](#)

[HEIGHT](#) *As used in <TABLE>*

[ROWSPAN](#)

[<TABLE>](#)

[<TD>](#)

[<TR>](#)

[VALIGN](#) *As used in <TD>*

[WIDTH](#) *As used in <TABLE>*

[WIDTH](#) *As used in <TD>*

Netscape's 216 colors

Compiled by Professional Web Design - <http://junior.apk.net/~jbarta/>

330000	333300	336600	339900	33CC00	33FF00	66FF00	66CC00	669900	666600	663300	660000	FF0000	FF3300	FF6600	FF9900	FFCC00	FFFF00
330033	333333	336633	339933	33CC33	33FF33	66FF33	66CC33	669933	666633	663333	660033	FF0033	FF3333	FF6633	FF9933	FFCC33	FFFF33
330066	333366	336666	339966	33CC66	33FF66	66FF66	66CC66	669966	666666	663366	660066	FF0066	FF3366	FF6666	FF9966	FFCC66	FFFF66
330099	333399	336699	339999	33CC99	33FF99	66FF99	66CC99	669999	666699	663399	660099	FF0099	FF3399	FF6699	FF9999	FFCC99	FFFF99
3300CC	3333CC	3366CC	3399CC	33CCCC	33FFCC	66FFCC	66CCCC	6699CC	6666CC	6633CC	6600CC	FF00CC	FF33CC	FF66CC	FF99CC	FFCCCC	FFFFCC
3300FF	3333FF	3366FF	3399FF	33CCFF	33FFFF	66FFFF	66CCFF	6699FF	6666FF	6633FF	6600FF	FF00FF	FF33FF	FF66FF	FF99FF	FFCCFF	FFFFFF
0000FF	0033FF	0066FF	0099FF	00CCFF	00FFFF	99FFFF	99CCFF	9999FF	9966FF	9933FF	9900FF	CC00FF	CC33FF	CC66FF	CC99FF	CCCCFF	CCFFFF
0000CC	0033CC	0066CC	0099CC	00CCCC	00FFCC	99FFCC	99CCCC	9999CC	9966CC	9933CC	9900CC	CC00CC	CC33CC	CC66CC	CC99CC	CCCCC	CCFFCC
000099	003399	006699	009999	00CC99	00FF99	99FF99	99CC99	999999	996699	993399	990099	CC0099	CC3399	CC6699	CC9999	CCCC99	CCFF99
000066	003366	006666	009966	00CC66	00FF66	99FF66	99CC66	999966	996666	993366	990066	CC0066	CC3366	CC6666	CC9966	CCCC66	CCFF66
000033	003333	006633	009933	00CC33	00FF33	99FF33	99CC33	999933	996633	993333	990033	CC0033	CC3333	CC6633	CC9933	CCCC33	CCFF33
000000	003300	006600	009900	00CC00	00FF00	99FF00	99CC00	999900	996600	993300	990000	CC0000	CC3300	CC6600	CC9900	CCCC00	CCFF00

These are the grays:

000000	333333	666666	999999	CCCCCC	FFFFFF

These are the 16 basic Windows colors:

CC0000	FF0000	CC00CC	FF00FF	0000CC	0000FF	00CCCC	00FFFF	00CC00	00FF00	CCCC00	FFFF00	000000	999999	CCCCCC	FFFFFF